Using Wavelet Network in Nonparametric Estimation

Qinghua Zhang

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Using Wavelet Network in Nonparametric Estimation

Qinghua Zhang

N° 2321

June, 1994

PROGRAMME 5

ISSN 0249-6399

Using Wavelet Network in Nonparametric Estimation*

Qinghua Zhang

Programme 5 — Traitement du signal, automatique et productique
Projet AS

Rapport de recherche n° 2321 — June, 1994 — 44 pages

Abstract: In this paper one approach is proposed for using wavelets in non parametric regression estimation. The proposed non parametric estimator, named *wavelet network*, has a neural network like structure, but consists of wavelets. It makes use of techniques of *regressor selection* completed with *backpropagation* procedures. It is capable of handling *nonlinear* regressions of moderately large input dimension with *sparse* training data. Numerical examples are reported to illustrate the performance of this proposed approach.

Key-words: non parametric regression, wavelets, nonlinear modelling, neural network.

(Résumé : tsvp)

*This work has been partially supported by Alcatel-Alsthom-Recherche and European Gas Turbine SA.

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

Utiliser le réseau d'ondelettes dans estimation non paramétrique

Résumé : Dans ce document une approche est proposée pour utiliser des ondelettes en estimation non paramétrique. L'estimateur proposé, appelé *réseau d'ondelettes*, a une structure similaire à celle de certains réseaux de neurones. Il utilise des techniques de la sélection de regresseurs complétées par des procédures dites "backpropagation". Il est capable de traiter des problèmes de regression non linéaire de dimensions modérées avec des données d'apprentissage creuses. Des exemples numériques sont présentés pour illustrer la performance de l'approche proposée.

Mots-clé : regression non paramétrique, ondelettes, modélisation non linéaire, réseau de neurones.

1 Introduction

The *wavelet theory* is a rapidly developing branch of mathematics which has found many applications, for instance, in numerical analysis and signal processing (see e.g., [1, 2]). Though this attractive theory has offered very efficient algorithms for analyzing, approximating, estimating and compressing functions or signals, the applications of such algorithms are usually limited to problems of small input dimension[1] d, typically with $d = 1$ or 2, because constructing and storing wavelet basis of large input dimension are of prohibitive cost [3]. In order to handle problems of large input dimension, it is desirable to find some technique whose complexity is less sensitive to the input dimension. It seems that (artificial) *neural networks* are promising candidates for such purpose. Many studies have been reported on using neural networks for approximating functions of possibly large input dimension [4, 5, 6, 7, 8]. The *neural networks* are a class of computational architectures that are composed of interconnected nodes (neurons). Its name reflects its initial inspiration from biological neural systems, though the functioning of today's artificial neural networks may be quite different from that of the biological ones. Although some theoretical results have shown the potential of using *feedforward neural networks* as universal approximators [4, 6], the implementation of such networks suffers from the lack of efficient constructive method, both for determining the parameters of the neurons and for choosing the network structure. Recently, due to the similarity between *discrete inverse wavelet transform* and *one-hidden-layer neural network*, the idea of combining both wavelets and neural networks has been proposed by Q. Zhang and A. Benveniste in [9], and further developed in subsequent works [10, 11]. In [9] the *wavelet network* was introduced that is a class of feedforward networks composed of wavelets. There are several advantages by combining wavelets and neural networks, for instance, results from the wavelet theory are a better alternative of the theorems on the approximation ability of the feedforward neural networks, and the wavelet transform provides useful guidelines for the construction of wavelet networks. Several authors have independently studied on the connection of the wavelet theory and neural networks. In [12] the discrete wavelet transform was used by Y.C. Pati and P.S. Krishnaprasad for analyzing and synthesizing feedforward neural networks; in [13] J. Hong used orthonormal wavelet bases for constructing wavelet based neural network; and similar works have been reported by B.R. Bakshi and G. Stephanopoulos in [14].

[1] By *input dimension* d we mean the dimension of the analyzed signal or the number of variables of the analyzed function. See Problem 1 in this introduction section.

The general purpose of this paper is to solve the problem of *nonparametric regression estimation* that we formulate below.

Problem 1 (nonparametric regression) *Consider the following nonparametric regression model involving an unknown function* $f : \mathbf{R}^d \to \mathbf{R}$*:*

$$Y = f(X) + e \tag{1}$$

where X *and* Y *are random variables with values in* $\mathbf{R}^d$ *and* $\mathbf{R}$ *respectively*[2]*, and* e *is some noise of zero mean and independent of* X*. The* regression function f *is assumed to be nonlinear, belonging to some functional space (continuous, square integrable, etc.).* d *is called the* input dimension *of the regression model. Let*

$$\mathcal{O}_1^N = \{(x_1, x_2), \ldots, (x_N, y_N)\}$$

be a sample of the input-output pair (X, Y) *that we refer to as the* training data *set. The problem is to find nonparametric estimators* $\widehat{f}$ *of* f *based on* $\mathcal{O}_1^N$*.*

Note that by "nonparametric" we mean that we do restrict $\widehat{f}$ to some parameterized class, however this class must be sufficiently flexible, and the number of parameters in $\widehat{f}$ is not *a priori* fixed. Two categories of approaches for solving Problem 1 are distinguished in the literature of nonparametric regressions:

1. the *deterministic design* for which the sampled inputs $x_1, \ldots, x_N$ in $\mathcal{O}_1^N$ are assumed to be given design points;

2. the *random design* for which $\mathcal{O}_1^N$ is assumed to be a random sample of the pair of random variables (X, Y), following some distribution, with $f(x) = \mathbf{E}(Y|X = x)$.

The approach presented in this paper falls into the first category, though for some of its applications, the training data $\mathcal{O}_1^N$ may actually result from some random realization. They are considered as nonrandom design points, whenever they are given.

Obviously nonparametric estimators as formulated in Problem 1 are useful for *black-box* nonlinear system modelling. The design of such an estimator consists of two steps: choosing the class of $\widehat{f}$ and developing an algorithm for determining the parameters of $\widehat{f}$ based on the training data $\mathcal{O}_1^N$. The wavelet networks are chosen

[2] For the sake of convenience, we refer to X and Y as the *input* and the *output* respectively, though they do not necessarily need to be so in actual applications.

as the class of $\widehat{f}$ in our proposed approach. And one algorithm for determining the parameters of $\widehat{f}$, that is the procedure for building wavelet network based on training data, will be proposed in this paper by combining techniques in *regression analysis* and the commonly used *backpropagation* procedures.

In order to evaluate the performance of nonparametric estimators, or at least to compare different nonparametric estimators, it is necessary to introduce some figures of merit. For theoretical analysis, this is usually accomplished via some norm of the estimation error $\|f - \widehat{f}\|$, for instance, the L_p-norms. However, for a given nonparametric regression problem as defined in (1), apart from some assumptions on the regularity of f (e.g. belonging to some functional space), the only available information about f is given by the training data $\mathcal{O}_1^N$, possibly coming with some assumptions on the statistical property of the measurement noise e. Therefore, in order to evaluate the performance of a chosen estimator $\widehat{f}$ on a given nonparametric regression problem, some criterion based on $\mathcal{O}_1^N$ should be defined in practice. For this purpose we adopt the commonly used mean of square errors

$$\text{MSE} = \frac{1}{N}\sum_{k=1}^{N}(\widehat{f}(x_k) - y_k)^2$$

as an evaluation of the estimator $\widehat{f}$. In order to balance between the estimation errors and the complexity of the estimator, in section 7 we shall consider some alternative of this criterion that will be useful for adaptively determining the number of wavelets in the estimator.

Usually, nonparametric estimators suffer from the curse of dimensionality, that is, the complexity of the estimators grows drastically with the input dimension d. This phenomenon is related to the fact that the sample length N is in the order of the exponential of d if the sampled inputs $\{x_1, \ldots, x_N\}$ fill a regular lattice of constant resolution in $\mathbf{R}^d$. In practice this curse of dimensionality can be avoided only in some particular situations where the estimated f has in some sense a small dimensional character. Below are some examples of such situations.

- The regression function f is essentially linear.
- The structure of the estimator $\widehat{f}$ coincides with the form of f. In other words, the chosen mathematical model coincides with the physical phenomenon under consideration.
- The components of the input X are weakly interacted.

- The regression function f is is smooth *almost* everywhere in its definition domain.
- The training data $\mathcal{O}_1^N$ are sparse in the input space $\mathbf{R}^d$ and only the parts of this space "explored" by the sparse training data are of interest. In other words, f is observed and estimated only on the same small portion of the input space.

We are particularly interested in the last situation that often occurs in classification problems and in the modeling of control systems. Indeed it is a reasonable situation in practice, because when the training data $\mathcal{O}_1^N$ are sparse, the sample length N is not strongly related with the input dimension d. Hence in this paper we make the assumption that *the training data are sparse* in the case of large input dimension. In the design of our algorithms, we shall explore the sparse nature of the training data in order to increase their computational efficiency, and thus, to handle problems of moderately large input dimensions.

The paper is organized as follows. In section 2 some basic concepts of the wavelet theory are shortly introduced. The wavelet network structure and an outline of the proposed procedure for wavelet network construction are stated in section 3. Sections 4 through 7 contain details of the proposed procedure. Some numerical examples are shown in section 8. Finally, some concluding remarks are given in section 9.

2 Continuous and discrete wavelet transforms

We shortly recall some basic concepts about wavelet transforms that will be useful for developing our nonparametric regression estimators.

2.1 The continuous wavelet transform

Historically the continuous wavelet transform was the first studied wavelet transform. Here we summarize some results concerning *radial wavelets* in $L_2(\mathbf{R}^d)$.

Definition 1 *A pair of radial functions $\varphi, \psi \in L_2(\mathbf{R}^d)$ are admissible as* analysis *and* synthesis wavelets*, if they satisfy the condition*

$$\int_0^\infty a^{-1} \widehat{\varphi}(a\omega)\widehat{\psi}(a\omega)\,\mathrm{d}a = 1, \quad \forall \omega \in \mathbf{R}^d \tag{2}$$

where $\widehat{\varphi}$ and $\widehat{\psi}$ are the Fourier transforms of φ and ψ, respectively.

Note that the integral in (2) does not depend on $\omega \neq 0$ since the functions φ and ψ are radial. In order for this integral to be well defined, it is sufficient that, for example, $\widehat{\varphi}(\omega)\widehat{\psi}(\omega) = O(\|\omega\|)$ [3]. Here $\|\omega\|$ denotes the Euclidean norm of ω. Once this integral is well defined and finite, a simple normalization will lead to a pair of φ, ψ that satisfies the condition (2). Here are two examples of such pairs [3]:

$$\psi(x) = \sqrt{2}(d - \|x\|^2)e^{-\frac{\|x\|^2}{2}} \ , \ \varphi(x) = \sqrt{2}e^{-\frac{\|x\|^2}{2}};$$
$$\psi(x) = \varphi(x) = \frac{1}{\sqrt{2}}(d - \|x\|^2)e^{-\frac{\|x\|^2}{2}}$$

Again $\|x\|$ denotes the Euclidean norm of x in the above equations.

In [15] the following theorem is proved.

Theorem 1 *Let φ and ψ be a pair of radial functions satisfying condition (2), then for any function $f \in L_2(\mathbf{R}^d)$, the following formulae define an isometry between $L_2(\mathbf{R}^d)$ and a subspace of $L_2(\mathbf{R}_+ \times \mathbf{R}^d)$:*

$$u(a,t) = a^{d-1/2} \int_{\mathrm{R}^d} f(x)\ \varphi(a(x-t))\ \mathrm{d}x \tag{3}$$

$$f(x) = \int_{\mathrm{R}_+ \times \mathrm{R}^d} u(a,t)\ \psi(a(x-t))\ a^{d-1/2}\ \mathrm{d}a\mathrm{d}t \tag{4}$$

$a \in \mathbf{R}_+$ and $t \in \mathbf{R}^d$ are respectively called *dilation* and *translation parameters.* Equations (3) and (4) define the (continuous) *wavelet transform* of f and its *inverse transform*, respectively. The continuous wavelet transform and its inverse transform are not directly implementable on digital computers. In practice, they have to be discretized. Several possibilities exist for this purpose, that we discuss below.

2.2 Wavelet bases and frames

When the inverse wavelet transform (4) is discretized into

$$f(x) = \sum_i u_i\ \psi(a_i x - t_i) \tag{5}$$

some conditions are required so that this discrete version of the reconstruction of f can actually hold. This corresponds to the sampling problem in the case of Fourier transform, for which the Shannon theorem is well known (see e.g. [16]). For the wavelet transform, several solutions are known. The first one is looking for some

countable family of (a_i, t_i), so that the corresponding family of dilated and translated wavelets

$$\{a_i^{d/2}\psi(a_i x - t_i) : i \in \mathbf{Z}\} \tag{6}$$

constitutes an orthonormal basis of some functional space, e.g. $L_2(\mathbf{R}^d)$. Usually a regular lattice $\{(a_0^n, mt_0) : n \in \mathbf{Z}, m \in \mathbf{Z}^d\}$ is used for the discretization where the scalar parameters a_0 and t_0 define the step sizes of dilation and translation discretizations. Wavelet bases have many applications in signal processing and numerical analysis because they offer very efficient algorithms and provide more useful information than Fourier transform. It is not always possible to build orthonormal wavelet bases with any wavelet function ψ. Though there are some well developed techniques for constructing the wavelet function ψ and its associated orthonormal basis (see, for example, [15, 17]), the wavelet function ψ has to satisfy strong restrictions. These restrictions lead to conflicts between the regularity and the compactness of the wavelet function, both being desired properties. Alternatively, bi-orthonormal wavelet families offer similar algorithms while requiring less restrictions on the wavelet function [15]. Further more, if one gives up the idea that the discrete wavelet family (6) should be a basis of some considered functional space and requires only that (6) constitutes a *frame*, then much more freedom on the choice of ψ is gained. Frames were introduced by Duffin and Schaeffer in the context of nonharmonic Fourier series [18]. Informally speaking, a frame is a "redundant basis". The following is a formal definition of the frame in a general Hilbert space.

Definition 2 *A sequence $\{\psi_k : k \in \mathbf{Z}\}$ in a Hilbert space $\mathcal{H}$ is called a* frame *of $\mathcal{H}$ if there exist two constants $A > 0$ and $B < \infty$ so that, for all $f \in \mathcal{H}$, the following inequalities hold.*

$$A\|f\|^2 \leq \sum_{k\in\mathbf{Z}} |\langle f, \psi_k\rangle|^2 \leq B\|f\|^2$$

Under the frame condition, f can be recovered from $\langle f, \psi_k\rangle$, $k \in \mathbf{Z}$, via some iterative procedure [15, 19]. Hence, if family (6) constitutes a frame, then the discrete reconstruction formula (5) is ensured. If a regular lattice of wavelets is considered

$$\{\psi(a_0^n x - mt_0) : n \in \mathbf{Z}, m \in \mathbf{Z}^d\}\ , \tag{7}$$

the section 3.3.2 of [15] gives sufficient conditions on ψ, a_0, t_0 so that family (7) constitutes a frame of $L_2(\mathbf{R})$. In [10] this result is extended to the case of $L_2(\mathbf{R}^d)$. Using frame rather than orthonormal basis leaves much more freedom in the choice of the wavelet function ψ, but the computation of the reconstruction coefficients u_i in (5) becomes non trivial [15].

2.3 Adaptive discretization and wavelet network

Though the wavelet bases and frames have been developed with efficient numerical algorithms, their applications have been limited to problems of small input dimension. The main reason is that wavelet bases and frames are usually constructed with regularly dilated and translated wavelets, independent of the available measured information (the training data). In practice, constructing and storing such wavelet basis or frame of large input dimension are of prohibitive cost. While the curse of dimensionality generally exists for nonparametric estimation, we should be able to overcome this difficulty in some particular situations, for instance when the function f to be estimated is mostly smooth but has localized irregularities, or when the available training data are sparse. We can expect that the wavelet estimator will be more efficient if the wavelet "basis" is constructed according to the training data. This yields the idea of adaptive discretization of the continuous wavelet transform. More precisely, in order to obtain a discrete reconstruction (5), instead of using a fixed lattice of (a_i, t_i), we can adaptively determine the values of (a_i, t_i), according to the function f or the sampled data $\mathcal{O}_1^N$. In this way, all the parameters u_i, a_i, t_i of (5) are to be adapted, thus (5) is very similar to a one-hidden-layer feedforward neural network. Therefore, we call such adaptive discrete inverse wavelet transform *wavelet networks*. From this point of view, (5) can be constructed using techniques of neural networks. Usually, neural networks used as nonparametric regression estimators are first randomly initialized and then trained by a backpropagation procedure [8]. The random initialization makes such learning procedures very inefficient. In contrast, wavelet networks can be initialized with regular wavelet lattice, as proposed in [9]. Though this method is intuitively more reasonable than random initializations, it suffers from the curse of dimensionality while applied to problems of large input dimension. Some other solutions are possible, for example, in [20] a constructive algorithm was proposed based on the law of large numbers. The approach we propose in this paper combines techniques in regression analysis and backpropagation procedures.

Remark that the regular lattices of wavelet frames are particular cases of adaptive discretizations of the continuous inverse wavelet transform, consequently the discrete reconstruction formula (5) must hold for some properly adapted (a_i, t_i). As further expected, if the function f has some particular property of regularity, better results can be obtained, due to the additional flexibility of the adaptive wavelet family. Let us cite the the following theorem proved by B. Delyon *et al.* [20] in the Sobolev space.

Theorem 2 *ψ is any radial wavelet function such that there exists a related radial function φ which satisfies condition (2). Let p, μ, l, ρ be real numbers satisfying*

$$1 < p < \left(1 - \frac{\rho - l}{d}\right)^{-1}, \qquad \mu = \min\left(1 - \frac{1}{p}\ ,\ \frac{1}{2}\right)$$

and f be a function of the Sobolev space $\mathcal{W}_1^\rho(\mathbf{R}^d)$; then, for any $n > 0$ there exists a function f_n of the form

$$f_n(x) = \sum_{i=1}^{n} u_i\ \psi(a_i(x - t_i)) \tag{8}$$

such that

$$\|f_n - f\|_{\mathcal{W}_p^l} \leq C n^{-\mu} \|f\|_{\mathcal{W}_1^\rho}\ .$$

where $C < +\infty$ is a constant. In particular, if $\rho > d/2$ then

$$\|f_n - f\|_{L_2} \leq n^{-1/2} C \|f\|_{L_1}^\rho\ .$$

This theorem provides us with an upper bound for the rate of approximation in Sobolev space when adaptive dilation/translation discretization of the continuous wavelet transform is used.

Another idea for nonparametric estimation using a data dependent wavelet family is to start with an orthonormal wavelet basis and then to cut off from the basis the wavelets of small transform coefficients. This method is called by some authors *wavelet shrinkage* [3, 21]. Since it uses orthonormal wavelet bases, it is difficult to implement it in the case of large input dimension.

In the following sections, we explain how to efficiently implement the adaptive discrete inverse wavelet transform by means of the wavelet network.

3 The wavelet network and its structure

As explained in the previous section, a wavelet network is an adaptive discretization of the continuous inverse wavelet transform. It can also be considered as an one-hidden-layer neural network with wavelets as activation functions of its neurons. We refer the reader to [9] for heuristic comparisons between neural network and wavelet network.

Because large dimensional wavelets are much less studied than one dimensional wavelets, it is useful to introduce some simple ways for building multi-dimensional wavelets from one dimensional ones. The most natural ones are the tensor product form discussed in [9] and the radial form used in [10].

Given a wavelet function $\psi : \mathbf{R}^d \to \mathbf{R}$, its associated wavelet network is written as

$$f_n(x) = \sum_{i=1}^{n} u_i \psi(a_i \star (x - t_i)) \tag{9}$$

where $u_i \in \mathbf{R}, a_i \in \mathbf{R}^d, t_i \in \mathbf{R}^d$, and "$\star$" means component-wise product of two vectors. Note that we could have used scalar dilation parameters a_i, but we prefer the vectorial dilation parameters because they considerably increase the flexibility of network (9) at a reasonable price.

All the parameters a_i, t_i, u_i of network (9) are to be adapted on the training data. As for the neural networks, this is a difficult optimization problem. In principle we can initialize (9) with a regular wavelet lattice and apply backpropagation algorithms. However, by using a regular wavelet lattice, many useless wavelets may be included in the network. The idea for remedying this drawback is to apply techniques of regressor selection. The following is an outline of this proposed approach.

1. Construct a library W of discretely dilated and translated versions of a given wavelet ψ. This library is constructed according to the available training data set, typically by selecting a subset from some regular lattice of dilated and translated versions of ψ.

2. Not all wavelets from the library W are useful in fitting f from noisy data, however. Hence we are faced with the problem of selecting the best regressors from W, if the wavelets in W are considered as regressors. Three methods will be proposed for this. When the regressors are conveniently selected, fitting model (9) amounts to identifying the coefficients u_i, which is a standard least squares estimation problem.

3. Steps 1 and 2 above yield a fast training procedure. The result can still be further improved by subsequently applying an iterative backpropagation algorithm with steps 1 and 2 as its initialization. In fact, since the initialization was good, a faster quasi-Newton procedure should be used.

Details of each step are given in the following sections.

4 Constructing the wavelet library W

The wavelet library W is considered as a set of regressor candidates. It should contain a finite number of wavelets, as less as possible, so that the regressor selection procedures can be efficiently applied. Given a wavelet function ψ, the construction of W consists in selecting some subset of the continuously parameterized family

$$\{\psi(a(x-t)) : a \in \mathbf{R}_+, t \in \mathbf{R}^d\}.$$

This is in principle the same problem as discretizing the continuous wavelet reconstruction (4) to obtain the discrete reconstruction (5). The standard discretization is a regular lattice:

$$\{\psi(a_0^n x - m t_0)) : n \in \mathbf{Z}, m \in \mathbf{Z}^d\} \tag{10}$$

where $a_0, t_0 > 0$ are two scalar constants defining the discretization step sizes for dilation and translation, respectively. Typically we take a dyadic grid for the sake of simplicity.

Now the countable family (10) should be truncated into a finite set. Usually we only want to estimate $f(x)$ on a compact domain $D \subset \mathbf{R}^d$ and the wavelet function $\psi(x)$ is chosen to be compactly supported or rapidly vanishing. Hence we can replace in (10) $m \in \mathbf{Z}^d$ by $m \in S_t$ with a finite set $S_t \subset \mathbf{Z}^d$; on the other hand, $n \in \mathbf{Z}$ should be replaced by $n \in S_a$ with a finite set $S_a \subset \mathbf{Z}$ corresponding to the "desired" resolution levels of the estimation. In practice, 4 or 5 consecutive dilatation levels are usually sufficient, with the largest wavelet scale corresponding to the size of D. After such a truncation, family (10) becomes

$$\{\psi(a_0^n x - m t_0)) : n \in S_a, m \in S_t(n)\} \tag{11}$$

Because we are interested in estimating $f(x)$ with sparse data $\mathcal{O}_1^N$, it is possible that some (or even many) wavelets in (11) do not contain any sample point in their supports[3]. These wavelets are useless for processing the available data, so they should be removed from (11). Two methods are possible for this purpose:

1. scan all the wavelets in (11), for each of them examine if its support contains any sample points of the training data $\mathcal{O}_1^N$;

[3] If the wavelet is not compactly supported, but rapidly vanishing, the term *support* should be interpreted in an approximative way as some finite domain around the center of the wavelet. This comment is valid all over the rest of this paper.

2. scan the training data $\mathcal{O}_1^N$, for each sample point determine the wavelets in (11) whose supports contain the sample point.

For the first method, family (11) has to be actually built and then some of its elements are eliminated. If the input dimension d is large, it is difficult to handle family (11). For the second method, however, the input dimension d is not a critical factor of complexity, since family (11) does not need to be actually created. In practice the sample length N is limited, *this method allows to handle problems of relatively large input dimension.* In particular, if the supports of the wavelets are approximated by hyper-cubes in $\mathbf{R}^d$, the second method can be very easily implemented. This may be a rough approximation, but the purpose of this procedure is only to initialize wavelet networks which will be further trained.

From now on we denote by W the set of wavelets resulting from the refining of family (11) and call it the *library of wavelet regressor candidates* or *wavelet library* for short. For computational convenience we normalize the wavelets:

$$W = \left\{ \psi_i : \ \psi_i(x) = \alpha_i \psi(a_i(x - t_i)), \ \alpha_i = \left(\sum_{k=1}^{N} [\psi(a_i(x_k - t_i))]^2 \right)^{-\frac{1}{2}}, \quad i = 1, \ldots, L \right\} \quad (12)$$

where $x_1, \ldots, x_N$ are the sampled inputs in $\mathcal{O}_1^N$, L is the number of wavelets in W and a_i, t_i correspond to the dilation and translation parameters a_0^n and $a_0^{-n} m t_0$ of the wavelet ψ_i. The numbering order with i is arbitrary. Note that the dilation parameters a_i in W are scalar, i.e., the wavelets are equally dilated along all the axes of $\mathbf{R}^d$.

5 Selecting best wavelet regressors

Selecting the "best" regressors from a finite set of regressor candidates is a classical problem in regression analysis [22]. In our case the set of regressor candidates is the wavelet library W. The problem is then to select a number $M \leq L$ of wavelets from W, the "best" ones based on the training data $\mathcal{O}_1^N$, for building the regression

$$f_M(x) = \sum_{i \in I} u_i \psi_i(x) \quad (13)$$

where I is a M-elements subset of the index set $\{1, 2, \ldots, L\}$, $u_i \in \mathbf{R}$.

Problem 2 (regressor selection) *Given a wavelet library W as defined in (12) and a set of training data $\mathcal{O}_1^N$, let $\mathcal{I}_M$ be the set of all the M-elements subsets of $\{1, 2, \ldots, L\}$, $M \leq L$. The problem is to find $I \in \mathcal{I}_M$ that minimizes*

$$J(I) = \min_{u_i, i \in I} \frac{1}{N} \sum_{k=1}^{N} \left(y_k - \sum_{i \in I} u_i \psi_i(x_k) \right)^2 .$$

Note that the minimization in u_i leads to the least squares solution with respect to u_i for any $I \in \mathcal{I}_M$.

In section 7 we shall deal with the problem of determining the value of M. For the moment we simply assume that the value of M is given. For convenience of presentation, let us define some vectorial notations.

$$v_i = \begin{bmatrix} \psi_i(x_1) \\ \vdots \\ \psi_i(x_N) \end{bmatrix}$$

where $\psi_i \in W$ and $x_1, \ldots, x_N$ are the sampled inputs in $\mathcal{O}_1^N$. Since ψ_i is normalized, v_i is unitary:

$$v_i^T v_i = 1, \quad i = 1, \ldots, L.$$

Now collect all the v_i, $i = 1, \ldots, L$ in a set V:

$$V = \{v_1, \ldots, v_L\}.$$

We also define the output sample vector

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \tag{14}$$

where $y_1, \ldots, y_N$ are the sampled outputs in $\mathcal{O}_1^N$.

Let $\text{span}(\{v_i : i \in I\})$ be the space linearly spanned by the vectors v_i, $i \in I$. With all these notations, Problem 2 is equivalent to selecting M vectors v_i from V that minimize the Euclidean distance from the vector y to the space $\text{span}(\{v_i : i \in I\})$. The Euclidean distance from a vector $y \in \mathbf{R}^N$ to a sub-space of $\mathbf{R}^N$ is defined as the Euclidean distance between y and its orthogonal projection onto the sub-space.

In principle such a selection can be performed by examining all the M-elements subsets of W, but the combination of all the possible subsets is usually very large and the exhaustive examination may not be feasible in practice. Some sub-optimal and heuristic solutions have to be considered. In the following we propose to apply several such heuristic procedures.

The residual based selection

The idea of this method is to select, for the first stage, the wavelet in W that best fits the training data $\mathcal{O}_1^N$, and then repeatedly select the wavelet that best fits the residual of the fitting of the previous stage. In the literature of the classical regression analysis, it is considered as a simple, but not effective method, for example in [22] where it is called *stage-wise regression procedure*. It is also similar to the *projection pursuit regression* (PPR) [23, 24], but much simpler than the latter. Because for classical regressions the number of regressor candidates is relatively small, some more complicated and more effective procedures are preferred. In our situation the number of regressor candidates may reach several hundreds or even more, the computational efficiency becomes more important and the simplicity of this method is of interest. Recently it is also used in some similar problems, such as the matching pursuit algorithm of S. Mallat and Z. Zhang [25] and the adaptive signal representation of S. Qian and D. Chen [26]. We describe in detail this procedure below.

Define the initial residual $\gamma_0(k) = y_k$, $k = 1, \ldots, N$, with y_k the output observations in $\mathcal{O}_1^N$. Set $f_0(x) \equiv 0$.

At stage i ($i = 1, \ldots, M$), search among W the wavelet ψ_j that minimizes

$$J(\psi_j) = \frac{1}{N} \sum_{k=1}^{N} (\gamma_{i-1}(k) - u_j \psi_j(x_k))^2$$

where

$$u_j = \left(\sum_{k=1}^{N} (\psi_j(x_k))^2 \right)^{-1} \sum_{k=1}^{N} \psi_j(x_k) \gamma_{i-1}(k).$$

and $\gamma_{i-1}(k)$ ($k = 1, \ldots, N$) are the residuals of stage $i-1$. Note

$$l_i = \arg \min_{1 \le j \le L} J(\psi_j)$$

then ψ_{l_i} is the wavelet selected at stage i. Update f_i and γ_i:

$$\begin{aligned} f_i(x) &= f_{i-1}(x) + u_{l_i} \psi_{l_i}(x) \\ \gamma_i(k) &= \gamma_{i-1}(k) - u_{l_i} \psi_{l_i}(x_k), \quad k = 1, \ldots, N. \end{aligned}$$

This procedure can be more conveniently described with the aid of vectorial notations as follows.

Define the initial residual vector $\gamma_0 = y$ with y as defined in (14) and set $f_0(x) \equiv 0$.

At stage i ($i = 1, \ldots, M$), search among V the vector v_j that minimizes

$$J(v_j) = (\gamma_{i-1} - u_j v_j)^T (\gamma_{i-1} - u_j v_j)$$

with

$$u_j = (v_j^T v_j)^{-1} v_j^T \gamma_{i-1} = v_j^T \gamma_{i-1}$$

where the last equality is due to the normality $v_j^T v_j = 1$.

Substituting u_j into $J(v_j)$ yields

$$\begin{aligned} J(v_j) &= (\gamma_{i-1} - v_j^T \gamma_{i-1} v_j)^T (\gamma_{i-1} - v_j^T \gamma_{i-1} v_j) && (15) \\ &= \gamma_{i-1}^T \gamma_{i-1} + (v_j^T \gamma_{i-1})^2 v_j^T v_j - 2(v_j^T \gamma_{i-1})^2 && (16) \\ &= \gamma_{i-1}^T \gamma_{i-1} - (v_j^T \gamma_{i-1})^2 && (17) \end{aligned}$$

It turns out that minimizing $J(v_j)$ at stage i is equivalent to maximizing $(v_j^T \gamma_{i-1})^2$.

The algorithm is summarized as follows.

Regressor Selection Algorithm 1

Step 0**:** Set $\gamma_0 = y$ and $f_0(x) \equiv 0$;

Step i **(**$i = 1, \ldots, M$**):** Let $I_i = \{j : j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{i-1}\}$, find

$$l_i = \arg \max_{j \in I_i} (v_j^T \gamma_{i-1})^2$$

and set

$$\begin{aligned} u_{l_i} &= v_{l_i}^T \gamma_{i-1} \\ f_i(x) &= f_{i-1}(x) + u_{l_i} \psi_{l_i}(x) \\ \gamma_i &= \gamma_{i-1} - u_{l_i} v_{l_i}. \end{aligned}$$

□

It is easy to prove (see [25]):

$$\gamma_i^T \gamma_i = \gamma_{i-1}^T \gamma_{i-1} - (v_{l_i}^T \gamma_{i-1})^2$$

so $\gamma_i^T \gamma_i$ monotonically decreases as i increases. It also means that the i-th term added to $f_M(x)$ has a contribution to the minimization of $\gamma_M^T \gamma_M$ measured by $(v_{l_i}^T \gamma_{i-1})^2$.

Stepwise selection by orthogonalization

The above residual based selection procedure does not explicitly consider the interaction or the non orthogonality of the wavelets in W. The idea of this alternative method is to select, for the first stage, the wavelet in W that best fits the training data $\mathcal{O}_1^N$, and then repeatedly select the wavelet that best fits $\mathcal{O}_1^N$ while working together with the previously selected wavelets. In terms of vectorial notations, this amounts to repeatedly selecting the $v_j \in V$ that linearly spans, with the previously selected vectors, the space the closest to y. Recently this method has been used in radial basis function (RBF) networks and other nonlinear modelling problems by S. Chen *et al.* [27, 28]. Here we present a slightly modified version of this algorithm with improved computational efficiency.

At stage i of this procedure, assume that the $i-1$ already selected wavelets correspond to the vectors $v_{l_1}, \ldots, v_{l_{i-1}}$. In order to select the i-th wavelet, we have to compute the distance from y to the space $\text{span}(v_{l_1}, \ldots, v_{l_{i-1}}, v_j)$ for each $j = 1, \ldots, L$ and $j \neq l_1, \ldots, l_{i-1}$. For computational efficiency, we orthogonalize the later selected vectors v_j to the earlier selected ones. Assume that $v_{l_1}, \ldots, v_{l_{i-1}}$ are already orthonormalized and renamed as $w_{l_1}, \ldots, w_{l_{i-1}}$, then $\text{span}(v_{l_1}, \ldots, v_{l_{i-1}}, v_j) = \text{span}(w_{l_1}, \ldots, w_{l_{i-1}}, v_j)$. For each $j = 1, \ldots, L$ and $j \neq l_1, \ldots, l_{i-1}$, compute

$$p_j = v_j - \left((v_j^T w_{l_1}) w_{l_1} + \cdots + (v_j^T w_{l_{i-1}}) w_{l_{i-1}} \right) \tag{18}$$

$$q_j = (p_j^T p_j)^{-\frac{1}{2}} p_j \tag{19}$$

then we should search the v_j, or equivalently the q_j, that minimizes

$$\begin{aligned} J(v_j) &= J(q_j) \\ &= \left[y - (\tilde{u}_{l_1} w_{l_1} + \cdots + \tilde{u}_{l_{i-1}} w_{l_{i-1}} + \tilde{u}_j q_j) \right]^T \left[y - (\tilde{u}_{l_1} w_{l_1} + \cdots + \tilde{u}_{l_{i-1}} w_{l_{i-1}} + \tilde{u}_j q_j) \right] \\ &= \left[y - W_j U_j \right]^T \left[y - W_j U_j \right] \end{aligned}$$

with the matrix $W_j = (w_{l_1}, \ldots, w_{l_{i-1}}, q_j)$ and the vector

$$U_j = (\tilde{u}_{l_1}, \ldots, \tilde{u}_{l_{i-1}}, \tilde{u}_j)^T = \left(W_j^T W_j \right)^{-1} W_j^T y = W_j^T y \tag{20}$$

where the last equality is due to the orthonormality of $w_{l_1}, \ldots, w_{l_{i-1}}, q_j$. Continue the computation:

$$\begin{aligned} J(v_j) &= y^T y + U_j^T W_j^T W_j U_j - 2 U_j^T W_j^T y \\ &= y^T y + U_j^T U_j - 2 U_j^T W_j^T y \end{aligned}$$

By (20) we have $U_j = W_j^T y$, therefore

$$\begin{aligned} J(v_j) &= y^T y + U_j^T U_j - 2U_j^T U_j \\ &= y^T y - U_j^T U_j \\ &= y^T y - \left(\tilde{u}_{l_1}^2 + \cdots + \tilde{u}_{l_{i-1}}^2 + \tilde{u}_j^2\right) \end{aligned}$$

Consequently minimizing $J(v_j)$ is equivalent to maximizing $\tilde{u}_{l_1}^2 + \cdots + \tilde{u}_{l_{i-1}}^2 + \tilde{u}_j^2$. By (20) we have

$$\begin{aligned} \tilde{u}_{l_k} &= w_{l_k}^T y, \quad k = 1, \ldots, i-1 \\ \tilde{u}_j &= q_j^T y \end{aligned}$$

so $\tilde{u}_{l_1}^2 + \cdots + \tilde{u}_{l_{i-1}}^2$ is independent of q_j. We conclude that minimizing $J(v_j)$ is equivalent to maximizing $\tilde{u}_j^2 = (q_j^T y)^2$.

After M iterations, the values of $l_1, \ldots, l_M$ are determined. Then

$$y = [w_{l_1}, \ldots, w_{l_M}][\tilde{u}_{l_1}, \ldots, \tilde{u}_{l_M}]^T + \gamma_M = [v_{l_1}, \ldots, v_{l_M}][u_{l_1}, \ldots, u_{l_M}]^T + \gamma_M$$

therefore

$$[w_{l_1}, \ldots, w_{l_M}][\tilde{u}_{l_1}, \ldots, \tilde{u}_{l_M}]^T = [v_{l_1}, \ldots, v_{l_M}][u_{l_1}, \ldots, u_{l_M}]^T \tag{21}$$

The wavelet network will be

$$f_M(x) = \sum_{i=1}^{M} u_{l_i} \psi_{l_i}(x)$$

so we have to determine the values of u_{l_i}.

Combining (18) and (19) with $j = l_i$ yields

$$\begin{aligned} v_{l_i} &= \left((v_{l_i}^T w_{l_1}) w_{l_1} + \cdots + (v_{l_i}^T w_{l_{i-1}}) w_{l_{i-1}}\right) + p_{l_i} \\ &= \left((v_{l_i}^T w_{l_1}) w_{l_1} + \cdots + (v_{l_i}^T w_{l_{i-1}}) w_{l_{i-1}}\right) + (p_{l_i}^T p_{l_i})^{\frac{1}{2}} q_{l_i} \\ &= \left((v_{l_i}^T w_{l_1}) w_{l_1} + \cdots + (v_{l_i}^T w_{l_{i-1}}) w_{l_{i-1}}\right) + (p_{l_i}^T p_{l_i})^{\frac{1}{2}} w_{l_i} \\ &= (\alpha_{1i} w_{l_1} + \cdots + \alpha_{i-1i} w_{l_{i-1}}) + \alpha_{ii} w_{l_i} \end{aligned}$$

with

$$\begin{aligned} \alpha_{ki} &= v_{l_i}^T w_{l_k}, \quad k = 1, \ldots, i-1 \\ \alpha_{ii} &= (p_{l_i}^T p_{l_i})^{\frac{1}{2}} \end{aligned}$$

Consequently

$$[w_{l_1}, \ldots, w_{l_M}]A = [v_{l_1}, \ldots, v_{l_M}] \tag{22}$$

where A is the triangular matrix

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \cdots & & \alpha_{1M} \\ 0 & \alpha_{22} & \alpha_{23} & \cdots & & \alpha_{2M} \\ 0 & 0 & \alpha_{33} & \cdots & & \alpha_{3M} \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & \alpha_{M-1M-1} & \alpha_{M-1M} \\ 0 & 0 & \cdots & & 0 & \alpha_{MM} \end{bmatrix}$$

Then, u_{l_i} can be obtained by solving the triangular system of equations obtained by combining (21) and (22):

$$A\,[u_{l_1}, \ldots, u_{l_M}]^T = [\tilde{u}_{l_1}, \ldots, \tilde{u}_{l_M}]^T. \tag{23}$$

Up to now we have followed [27, 28] in the description of this algorithm. Notice that in (18) the projections

$$(v_j^T w_{l_1})w_{l_1} + \cdots + (v_j^T w_{l_{i-2}})w_{l_{i-2}}$$

have already been calculated at the previous steps, only the last one $(v_j^T w_{l_{i-1}})w_{l_{i-1}}$ needs to be actually calculated at the current step. This allows to considerably improve the computational efficiency of the algorithm. The modified algorithm is summarized as follows.

Regressor Selection Algorithm 2

Step 1: set

$$I_1 = \{1, 2, \ldots, L\}$$

find

$$l_1 = \arg\max_{j \in I_1} \left(v_j^T y\right)^2$$

and set

$$\begin{aligned} \tilde{u}_{l_1} &= v_{l_1}^T y \\ w_{l_1} &= v_{l_1} \\ \alpha_{11} &= 1 \\ p_j^{(1)} &= v_j, \quad j = 1, \ldots, L,\ j \neq l_1 \end{aligned}$$

Step i $(i = 2, \ldots, M)$**:**

$$I_i = I_{i-1} - \{l_{i-1}\}$$

for each $j \in I_i$, compute

$$\begin{aligned} p_j^{(i)} &= p_j^{(i-1)} - \left(v_j^T w_{l_{i-1}}\right) w_{l_{i-1}} \\ I_i &= I_i - \{j : \ p_j^{(i)} = 0\} \end{aligned}$$

find

$$l_i = \arg\max_{j \in I_i} \frac{\left(\left(p_j^{(i)}\right)^T y\right)^2}{\left(p_j^{(i)}\right)^T p_j^{(i)}}$$

and set

$$\begin{aligned} w_{l_i} &= \left(\left(p_{l_i}^{(i)}\right)^T p_{l_i}^{(i)}\right)^{-\frac{1}{2}} p_{l_i}^{(i)} \\ \tilde{u}_{l_i} &= w_{l_i}^T y \\ \alpha_{ki} &= v_{l_i}^T w_{l_k}, \quad k = 1, \ldots, i-1 \\ \alpha_{ii} &= \left(\left(p_{l_i}^{(i)}\right)^T p_{l_i}^{(i)}\right)^{\frac{1}{2}} \end{aligned}$$

Step $M+1$**:** solve (23) to obtain u_{l_i}, $i = 1, \ldots, M$, and build

$$f_M(x) = \sum_{i=1}^{M} u_{l_i} \psi_{l_i}(x).$$

□

Backward elimination

In contrary to the previous methods, the backward elimination method starts by building the regression (13) with all the wavelets of W, then eliminates one wavelet per stage, while trying to increase as less as possible the residual at each stage.

The regression with all the wavelets of W is written as

$$f_L(x) = \sum_{i=1}^{L} u_i \psi_i(x)$$

where u_i are determined by the least squares solution:

$$(u_1, \ldots, u_L)^T = \left[(v_1, \ldots, v_L)^T (v_1, \ldots, v_L)\right]^{-1} (v_1, \ldots, v_L)^T y. \tag{24}$$

Note that inverting the matrix $(v_1, \ldots, v_L)^T (v_1, \ldots, v_L)$ will cause problem if it is singular. This situation rarely occurs with the set V of vectors corresponding to the wavelet library W. Whenever it happens, the backward elimination algorithm does not work, the regressor selection algorithms 1 or 2 should be used instead.

The residuals

$$\gamma_L(k) = y_k - f_L(x_k), \quad k = 1, \ldots, N$$

can be written in its vectorial form as:

$$\gamma_L = y - (v_1, \ldots, v_L)(u_1, \ldots, u_L)^T. \tag{25}$$

Combining (24) and (25) we get

$$\gamma_L^T \gamma_L = y^T y^T - y^T V_0 (V_0^T V_0)^{-1} V_0^T y$$

where the matrix $V_0 = (v_1, \ldots, v_L)$.

If we remove one wavelet, say ψ_j from $f_L(x)$, the same computation can be repeated to get a similar result

$$\gamma_{L-1}^T \gamma_{L-1} = y^T y^T - y^T C(v_j|V_0) \left(C(v_j|V_0)^T C(v_j|V_0)\right)^{-1} C(v_j|V_0)^T y$$

where the operator C means the complement of a matrix, i.e., if a matrix $U = [U_1, U_2, U_3]$, then $C(U_2|U) = [U_1, U_3]$. Hence, the increment of the sum of square residual caused by removing ψ_j from $f_L(x)$ is

$$\begin{aligned} J(\psi_j) &= \gamma_{L-1}^T \gamma_{L-1} - \gamma_L^T \gamma_L \\ &= y^T V_0 (V_0^T V_0)^{-1} V_0^T y - y^T C(v_j|V_0) \left(C(v_j|V_0)^T C(v_j|V_0)\right)^{-1} C(v_j|V_0)^T y \end{aligned} \tag{26}$$

Removing from $f_L(x)$ the wavelet ψ_j that minimizes (26) yields $f_{L-1}(x)$. Repeat the same procedure to remove another wavelet from $f_{L-1}(x)$, and so on. This results in the following algorithm.

Regressor Selection Algorithm 3'

Step 0**:** set $V_0 = (v_1, \ldots, v_L)$;

Step i $(i = 1, \ldots, L-M)$**:** let $I_i = \{j : \ j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{i-1}\}$, find

$$l_i = \arg\max_{j \in I_i} y^T C(v_j|V_{i-1}) \left(C(v_j|V_{i-1})^T C(v_j|V_{i-1}) \right)^{-1} C(v_j|V_{i-1})^T y$$

set $V_i = C(v_{l_i}|V_{i-1})$;

Step $L-M+1$**:** let $I_{L-M+1} = \{j : \ j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{L-M}\}$, build

$$f_M(x) = \sum_{j \in I_{L-M+1}} u_j \psi_j(x)$$

with u_j the components of the vector u given by

$$u = \left(V_{L-M}^T V_{L-M} \right)^{-1} V_{L-M}^T \, y.$$

□

The computation required by this procedure is quite heavy. For instance $L-i+1$ matrices need to be inverted at step i. The computation for inverting the matrices can be reduced in the following way.

For any matrix $U = [U_1, U_2, U_3]$ where U_1, U_2, U_3 are sub-blocs of U,

$$U^T U = \begin{bmatrix} U_1^T U_1 & U_1^T U_2 & U_1^T U_3 \\ U_2^T U_1 & U_2^T U_2 & U_2^T U_3 \\ U_3^T U_1 & U_3^T U_2 & U_3^T U_3 \end{bmatrix}.$$

Assume $(U^T U)^{-1}$ is already calculated and partitioned in the same way as $U^T U$:

$$\left(U^T U \right)^{-1} = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & \Lambda_{13} \\ \Lambda_{21} & \Lambda_{22} & \Lambda_{23} \\ \Lambda_{31} & \Lambda_{32} & \Lambda_{33} \end{bmatrix}$$

Then the following formula can be verified by simple computations:

$$\begin{aligned} \left([U_1, U_3]^T [U_1, U_3] \right)^{-1} &= \begin{bmatrix} U_1^T U_1 & U_1^T U_3 \\ U_3^T U_1 & U_3^T U_3 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \Lambda_{11} & \Lambda_{13} \\ \Lambda_{31} & \Lambda_{33} \end{bmatrix} - \Lambda_{22}^{-1} \begin{bmatrix} \Lambda_{12} \\ \Lambda_{32} \end{bmatrix} \begin{bmatrix} \Lambda_{21} & \Lambda_{23} \end{bmatrix} \end{aligned} \quad (27)$$

In this way only $V_0^T V_0$ needs to be actually inverted with conventional method. Using (27), $\left(C(v_j|V_i)^T C(v_j|V_i)\right)^{-1}$ can be obtained from sub-blocs of $\left(V_i^T V_i\right)^{-1}$.

Algorithm 3' can be further simplified as follows.

Assume that $f_L(x)$ is built with all the wavelets of W as before. Now eliminate one wavelet from $f_L(x)$, say ψ_j, but keep the values of u_l unchanged, $l = 1, \ldots, L$, the residual becomes

$$\gamma_{L-1}(k) = y_k - (f_L(x_k) - u_j \psi_j(x_k)) = \gamma_L(k) + u_j \psi_j(x_k), \quad k = 1, \ldots, N.$$

so

$$\gamma_{L-1} = \gamma_L + u_j v_j$$

Then

$$\begin{aligned} \gamma_{L-1}^T \gamma_{L-1} &= \gamma_L^T \gamma_L + u_j^2 v_j^T v_j + 2 u_j \gamma_L^T v_j \\ &= \gamma_L^T \gamma_L + u_j^2 + 2 u_j \gamma_L^T v_j \end{aligned}$$

The last term of this equation can be neglected under the assumptions that γ_L is close to zero mean and independent of v_j. Therefore

$$\gamma_{L-1}^T \gamma_{L-1} - \gamma_L^T \gamma_L \approx u_j^2$$

This means that removing ψ_j from $f_L(x)$ will cause an increment of the sum of square residuals approximatively equal to u_j^2. Repeating the same reasoning on $f_{L-1}(x), f_{L-2}(x)$, etc. yields the following procedure.

Regressor Selection Algorithm 3

Step 0: set $V_0 = (v_1, \ldots, v_L)$;

Step i $(i = 1, \ldots, L-M)$**:** let $I_i = \{j : \ j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{i-1}\}$ and compute

$$u = \left(V_{i-1}^T V_{i-1}\right)^{-1} V_{i-1}^T y$$

where u is a vector composed of u_j, $j \in I_i$;
find

$$l_i = \arg\min_{j \in I_i} u_j^2$$

set $V_i = C(v_j|V_{i-1})$;

Step $L-M+1$**:** let $I_{L-M+1} = \{j : j = 1, \ldots, L \text{ and } j \neq l_1, \ldots, l_{L-M}\}$, build

$$f_M(x) = \sum_{j \in I_{L-M+1}} u_j \psi_j(x)$$

with u_j the components of the vector u given by

$$u = \left(V_{L-M}^T V_{L-M}\right)^{-1} V_{L-M}^T \, y.$$

□

Note that equation (27) is used for inverting $V_i^T V_i$, $i > 0$, only $V_0^T V_0$ is inverted with conventional algorithm. Alternatively, if the mother wavelet function ψ is chosen to have compact support, then the matrices V_i and $V_i^T V_i$ are sparse, $V_i^T V_i$ is symmetric and usually has diagonal dominance. In such situations, and for large matrices $V_i^T V_i$, instead of directly computing

$$u = \left(V_i^T V_i\right)^{-1} V_i^T y$$

iterative methods [29] can be used for solving

$$\left(V_i^T V_i\right) u = V_i^T y \, .$$

Discussion. Let us examine the computational burden of Algorithm 1, 2 and 3. The numbers of arithmetic operations at step i of each algorithm are summarized in table 1. In addition to these arithmetic operations, a search for the maximal component of a vector is performed at each step of the algorithms. Notice that the computations mainly consist of multiplications. Roughly speaking, Algorithm 2 requires about 4 times of computations that Algorithm 1 does. At step 0 of Algorithm 3, a $L \times L$ matrix needs to be inverted. If $M < L/2$, more steps are performed by Algorithm 3 than by the other algorithms; the contrary is true if $M > L/2$. In most cases Algorithm 2 has a good trade-off between the effectiveness of the regressor selection and the computational burden, and thus is the most recommended one. Nevertheless, since these algorithms are all heuristic, we cannot say one is always more effective than others. So, for a given problem, if one algorithm does not give satisfactory result, another one may probably do.

	Algorithm 1	Algorithm 2	Algorithm 3
addition	$(L-i+2)(N-1)$	$(3L-2i+3)(N-1)$	$(L-i)(L-i+N-2)$
subtraction	N	$(L-i+1)N$	$(L-i)^2$
multiplication	$(L-i+1)(N+1)+2N$	$(L-i+1)(4N+1)+iN$	$(L-i)(2L-2i+N)$
division	——	$L-i+N+1$	$L-i$
square root	——	1	——

Table 1: Numbers of arithmetic operations at step i.
L is the number of wavelets in the library W, N is the sample length of the training data $\mathcal{O}_1^N$.

6 Combining regressor selection and backpropagation

In the previous two sections an approach is presented for quick construction of wavelet network (9). We can consider this quick construction as the initialization of a backpropagation procedure that will further refine the wavelet network by adapting its dilation, translation and linear parameters on the training data. Note that in (9) we use vectorial dilation parameters a_i, but in the wavelet library W the dilation parameters are scalar. Before applying any backpropagation procedure, change the scalar dilation parameters resulting from the regressor selection procedures into vectors of identical components.

Here we do not want to repeat the description of backpropagation procedures that have been published in many papers and books on neural networks. We just emphasize that a quasi-Newton algorithm is preferred for training the wavelet network, due to the good performance of its initialization procedures.

This proposed method allows to estimate nonlinear regression functions as formulated in Problem 1. In order to more easily capture linear properties in regressions, in practice we modify (9) into

$$f_n(x) = \sum_{i=1}^{n} u_i \psi(a_i \star (x - t_i)) + c^T x + b \tag{28}$$

with a vector of direct linear connection parameters $c \in \mathbf{R}^d$ and a bias parameter $b \in \mathbf{R}$. The initialization procedures are slightly modified so that, when algorithm 1 or 2 is used, the terms b and $c^T x$ are always first selected; when algorithm 3 or 3' is used, b and $c^T x$ are never eliminated.

7 Choosing the number of wavelets

In the previous sections, when describing the regressor selection algorithms, we assumed that the number M of wavelets to be selected for constructing the wavelet network was given. Choosing this number is an important and difficult problem that is related to model structure or model order determination problem. Even for linear models, choosing model order is not a trivial problem. Several approaches exist, such as *generalized cross validation, model complexity penalty, statistical hypothesis test* (see [30] chapter 16) and *minimum description length criterion* [31]. Usually, both generalized cross validation and minimum description length criterion lead to the form of criterions with a term measuring the modelling error and a term corresponding to some model complexity penalty. We have numerically tested several criterions of this type for determining the number of wavelets in wavelet network. It turns out that the *Akaike's final prediction error criterion* (FPE) works quite well when tested on a variety of examples. This criterion is written as follows [30, 32]:

$$J_{\text{FPE}}(\widehat{f}) = \frac{1 + n_p/N}{1 - n_p/N} \frac{1}{2N} \sum_{k=1}^{N} (\widehat{f}(x_k) - y_k)^2$$

where n_p is the number of parameters in the estimator, $(x_k, y_k) \in \mathcal{O}_1^N$ are training data and N is the sample length of the training data. Because scalar dilation parameters are used for initialization, each wavelet has 1 dilation, d translation and 1 linear parameters. In addition, the network has $d + 1$ direct linear connections. Hence, the number of parameters of an initialized wavelet network is given by

$$n_p = M(d + 2) + d + 1$$

where M is the number of wavelets in the network, d is the input dimension. Note that $J_{\text{FPE}}(\widehat{f})$ is nothing but the mean of square errors (MSE) of the estimator multiplied by a factor of penalty in terms of the number of parameters.

The following procedure is proposed. When Algorithm 1 or 2 is used, after the selection of each wavelet, $J_{\text{FPE}}(\widehat{f})$ is evaluated; the procedure is stopped when the number of parameters of the selected wavelets is greater than or equal to the sample length N of the training data, or when all the wavelets in the library have been selected. When Algorithm 3 or 3' is applied, evaluate $J_{\text{FPE}}(\widehat{f})$ after the elimination of each wavelet; the elimination stops when only one wavelet remains in the network. In any case choose the number M of wavelets that minimizes the criterion $J_{\text{FPE}}(\widehat{f})$.

8 Numerical examples

The entire procedure for constructing wavelet network as a nonparametric regression estimator, including the regressor selection algorithms 1, 2 and 3, has been implemented in Matlab 4.1 language. Algorithm 3' is not implemented because of its high computational cost. The package of these programs is available via anonymous FTP [33]. In this section we present two numerical examples using the above package. For these examples, the wavelet function ψ has been chosen as $\psi(x) = (d - \|x\|^2)e^{-\frac{\|x\|^2}{2}}$, $\|x\|^2 = x^T x$, the so-called "Mexican hat".

8.1 One dimensional interpolating

This example is chosen to visually illustrate the performance of the wavelet network in nonlinear nonparametric estimation. The problem is very simple: interpolate the points depicted in figure 2. Assume these points were drawn according to some regression equation $y = f(x) + e$, the problem becomes to estimate the regression function f. First we have to determine how many wavelets should be used for estimating f. We simply apply the Akaike's final prediction error criterion that is depicted in figure 3. According to this criterion, 9 wavelets are an appropriate choice. Next, the initialization algorithms proposed in the previous sections are applied. Finally the initialized networks are trained by a Gauss-Newton procedure. The results of interpolation are shown in figures 4, 5 and 6 where we also indicated the time of computation (on a Sparc-2 Work Station) and the Matlab Flops measuring the computational burden. We can appreciate the quality of the interpolations initialized with Algorithms 2 and 3, and the speed of the algorithms.

8.2 Application to the modeling of a gas turbine system

In this subsection we present an application of the wavelet network to the modeling of a gas turbine system. Gas turbines are power motors, typically used in electrical power generators and aircrafts. Usually a gas turbine system is mainly composed of a compressor, one or more combustion chambers and a turbine, as illustrated in figure 1. The compressor produces high pressure air which is then mixed with the fuel. This mixed gas is burned in the combustion chambers to increase its temperature and pressure. The burned gas is then forwarded into the turbine. Its pressure drives the rotor of the turbine. Finally the gas is rejected at the exhaust of the turbine.

One of the purposes of our joint study with European Gas Turbine SA, Belfort, and Alcatel-Alsthom-Recherche, Marcoussis, was to develop a monitoring and diag-

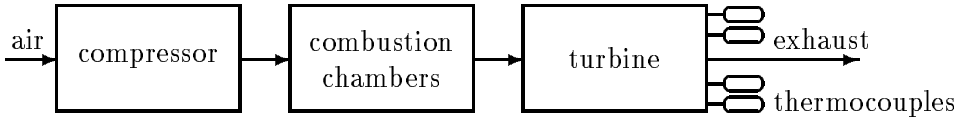

Figure 1: A gas turbine system

nostics system for the combustion chambers and the turbine, based on the measured pressure in the compressor, the rotation velocity of the turbine and measurements from the thermocouples installed at the exhaust of the turbine. A semi-physical model has been developed that predicts the profile of temperature at the exhaust of the turbine using the pressure in the compressor, the mean temperature at the exhaust of the turbine, and the rotation velocity of the turbine [34, 35]. Despite the inaccurate nature of this semi-physical model, it has been successfully used for developing a monitoring system of the combustion chambers, see [35]. Unfortunately, this model is not entirely satisfactory for some other purposes, such as the monitoring of the thermocouples installed at the exhaust of the turbine. Hence the wavelet network is applied to modeling this system.

Suggested by the semi-physical model, we assume that the temperature measured by each of the thermocouples installed at the exhaust of the turbine is a function of the mean temperature at the exhaust of the turbine (T_s), the compression rate π of the compressor, and the rotation velocity of the turbine (N). T_s, π and N are all measured. Therefore, we can try to construct, for each of the thermocouples, a wavelet network with T_s, π and N as its input variables, and train it to predict the temperature measured by the thermocouple.

We have tested this approach on the data taken from a gas turbine of European Gas Turbine SA. The training data were collected during about 48 hours. We have re-sampled the data and kept only 1000 measurement points. This gas turbine system is equipped with 18 thermocouples at its exhaust. For the sake of briefness, we show only the results concerning the first thermocouple. The re-sampled data are depicted in figure 7 where the plots correspond to T_s, π, N and $y = t_1 - T_s$ with t_1 the measurement of the first thermocouple. These 1000 measurement points, that we refer to as the *training data*, are used for training wavelet networks whose input vector is $x = (T_s, \pi, N)^T$. The obtained model will be tested on another set of measured data, that we refer to as the *test data* set and depict in figure 8. Testing identified models on a set of "fresh" data is called *cross-validation*. It is a commonly used method for evaluating the performance of identified models.

In order to choose the number wavelets of the wavelet network, the Akaike's final prediction error (FPE) criterion was applied. In figure 9 this criterion is plotted as a function of the number of wavelets used in the network. 40 wavelets is a suitable choice according to this criterion.

We initialize the wavelet networks with each of the proposed algorithms and train them with the Gauss-Newton procedure. In order to get some idea on the performance of the resulting models, we compare their results with those of the semi-physical model and a third degree polynomial model. In figures 10 and 11 are shown the results obtained with the semi-physical model and the third degree polynomial model, respectively. The results obtained with the wavelet networks initialized with Algorithm 1, 2 and 3, and the results after 10 iterations of the Gauss-Newton procedure are given in figures 12, 13 and 14. In table 2 we listed the mean of square errors (MSE) of all these models on the training data set as well as on the test data set. Networks 1, 2 and 3 are the networks initialized with Algorithm 1, 2, and 3, respectively. For each of these networks we give the result of its initialization (init. MSE) and the result after 10 iterations of the Gauss-Newton procedure (final MSE). The time of computation for building these models is also listed in table 2, based on our programs in Matlab 4.1 language executed on a Sun Sparc-2 Work Station, as well as the Matlab's Flop that measures the computational burden.

By examining these results, we can make the following observations:

- The semi-physical model performs quite poorly in predicting the output of the system.
- The system is actually nonlinear, as indicated by the results obtained with the polynomial model.
- The wavelet networks do improve the performance on prediction, but at the price of increasing computational complexity and losing the physical meaning of the model parameters.

9 Conclusion

We have presented one approach for nonparametric regression estimation using wavelet networks. In principle this proposed approach is by no means restricted to the wavelet networks. However, using wavelets as regressors have several advantages, for instance the local property of the wavelets makes efficient the estimation of regression functions having localized regularities, and the discrete wavelet transforms

models	network 1	network 2	network 3	semi-physical	polynomial
train. init. MSE	1.2656	1.0224	1.0381	—	—
train. final MSE	0.5395	0.4250	0.4503	3.5268	2.8438
test. init. MSE	1.2368	1.1229	1.1576	—	—
test. final MSE	1.1886	1.2348	1.0898	2.8914	2.1135
init. flops	2.0718×10^7	6.3663×10^7	7.5143×10^7	—	—
train. flops	1.5365×10^9	1.5478×10^9	1.5365×10^9	9.8041×10^8	4.7056×10^5
init. time (sec.)	41.6	105.5	87.2	—	—
train. time (sec.)	2461.8	2176.7	2456.5	2265.0	1.5362

Table 2: Performance evaluation of the models of the gas turbine
Networks 1, 2 and 3 are the wavelet networks initialized with Algorithm 1, 2, and 3, respectively.

provide useful guidelines for building the library of regressor candidates. The use of adaptively dilated and translated wavelet family allows to remedy the curse of dimensionality in the case of sparse training data. We have only discussed single output regressions, but the proposed approach is readily extended to multi-output case. These algorithms are mostly heuristic, more theoretic investigations are still needed in order to better understand and improve them. Another problem is that the radial (or elliptic) form of the wavelets makes all the input variables of the regression involved in all the wavelets. It is also to be improved in future works.

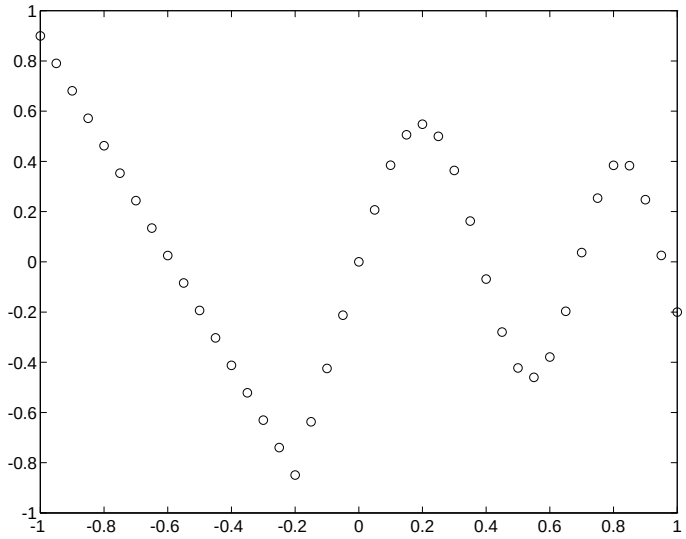

Figure 2: The data to be interpolated

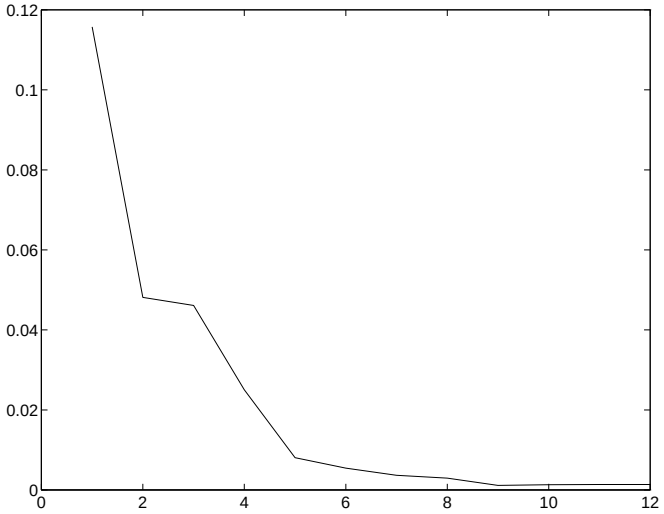

Figure 3: Akaike's final prediction error (FPE) criterion
Abscissa: number of wavelets, Ordinate: J_{FPE}

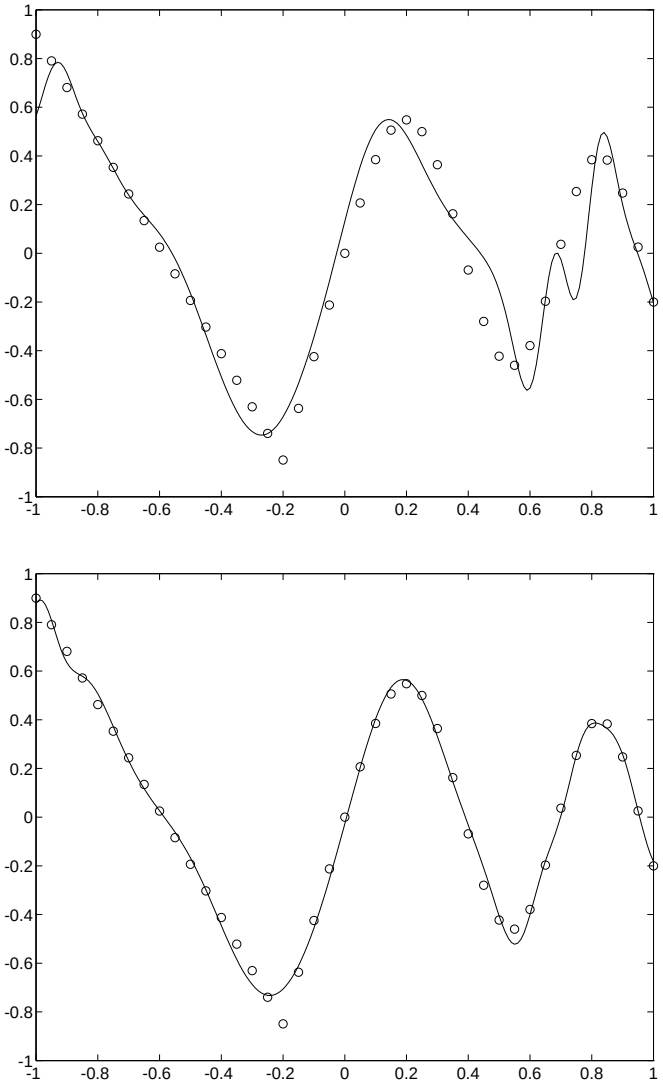

Figure 4: Interpolation initialized with Algorithm 1 (top) and trained 4 iterations (bottom)
Initialization: 1.41 seconds, 58865 flops
Training: 10.37 seconds, 513319 flops

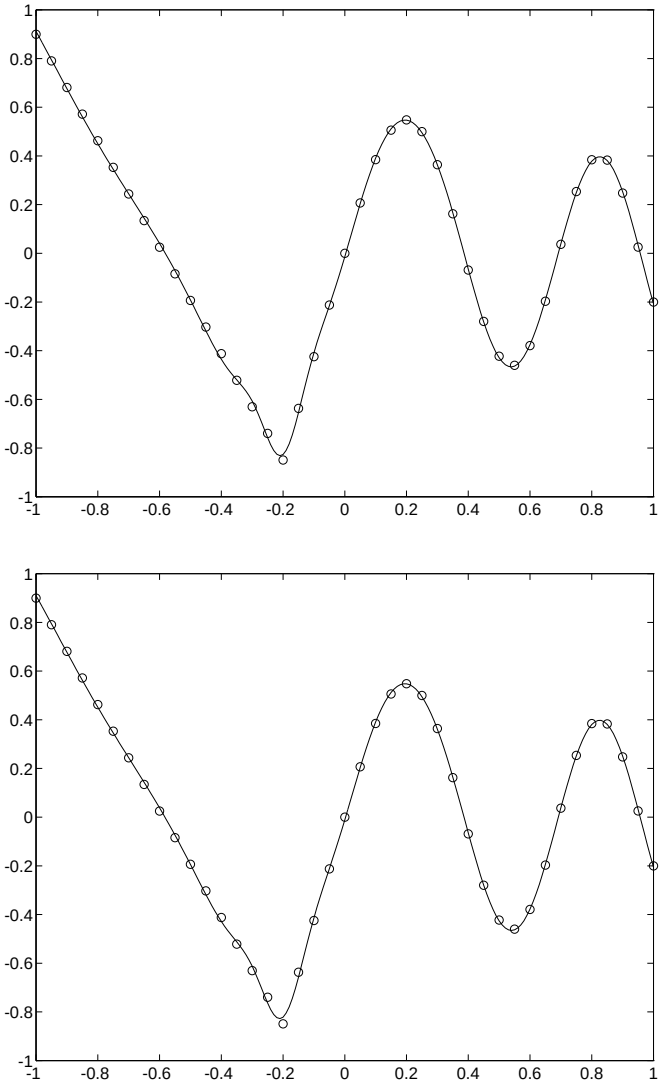

Figure 5:
Interpolation initialized with Algorithm 2 (top) and trained 4 iterations (bottom)
Initialization: 1.73 seconds, 137201 flops
Training: 7.96 seconds, 223066 flops

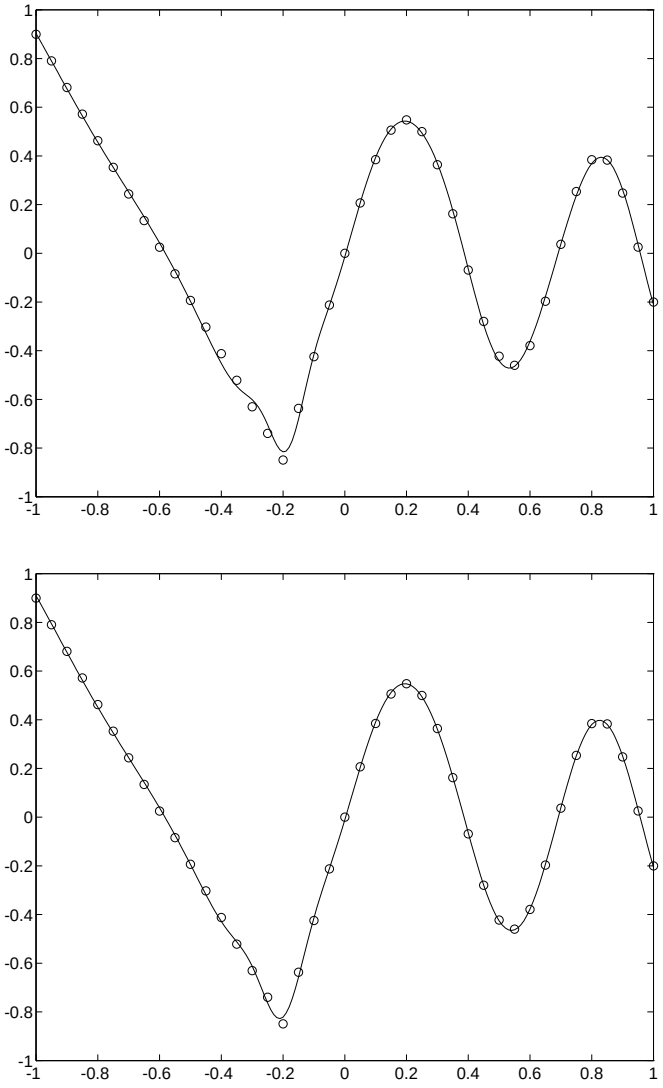

Figure 6:
Interpolation initialized with Algorithm 3 (top) and trained 4 iterations (bottom)
Initialization: 6.14 seconds, 357199 flops
Training: 8.33 seconds, 223071 flops

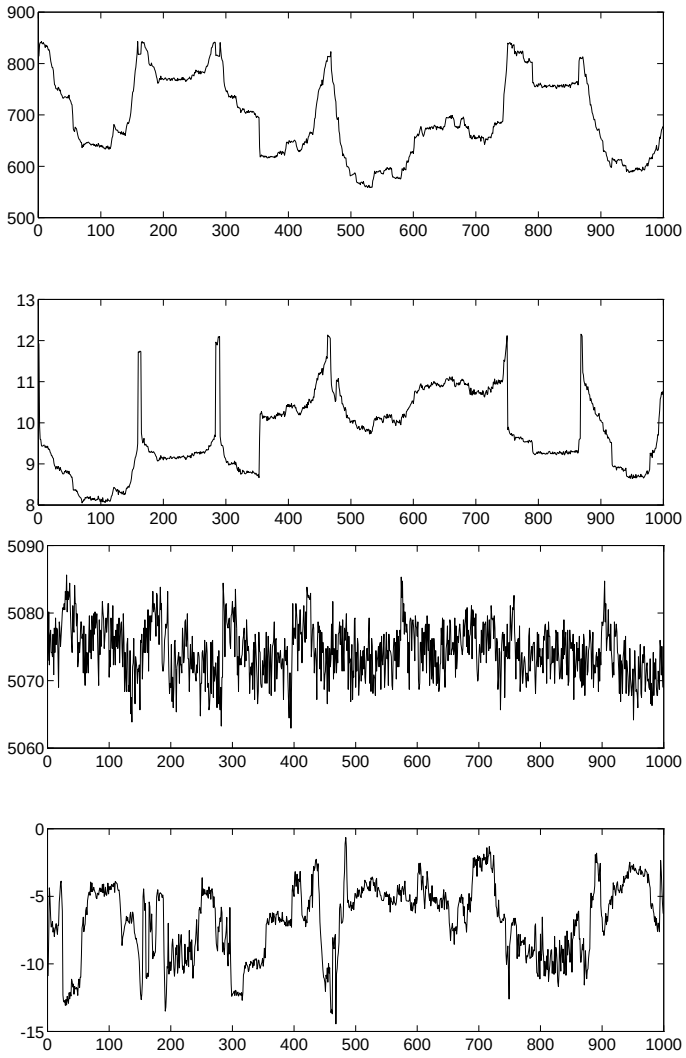

Figure 7: Training data. The plots correspond to, from top to bottom, T_s, π, N and $y = t_1 - T_s$.

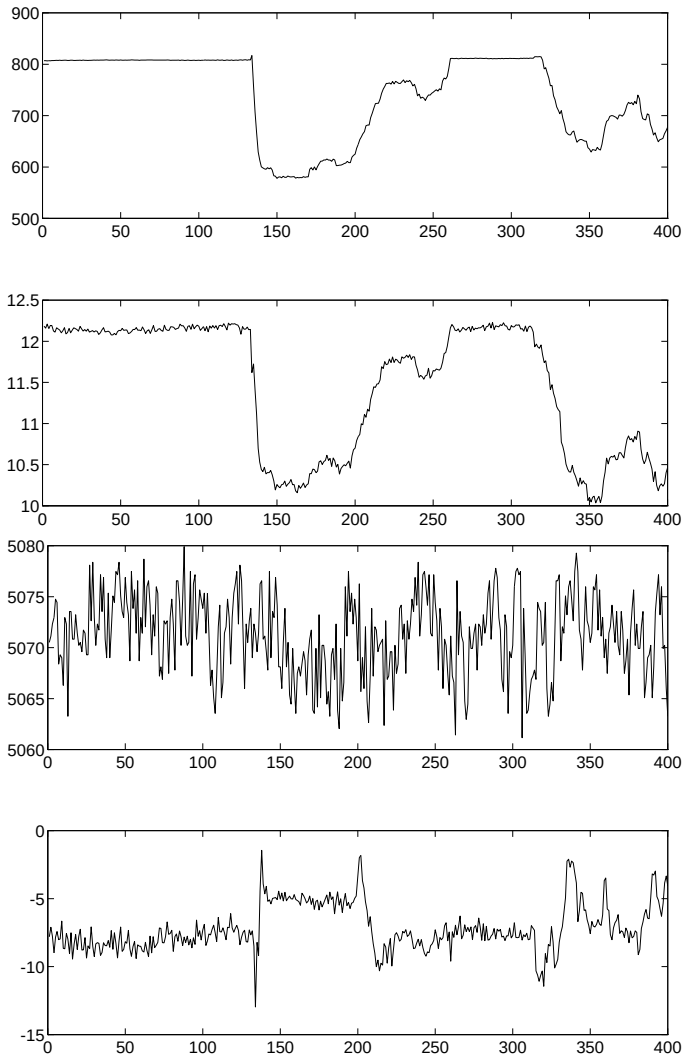

Figure 8: Test data. The plots correspond to, from top to bottom, T_s, π, N and $y = t_1 - T_s$.

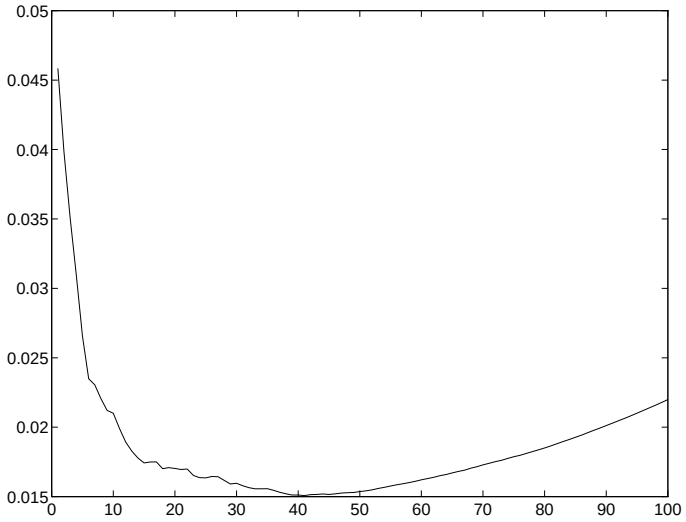

Figure 9: Akaike's final prediction error (FPE) criterion
Abscissa: number of wavelets, Ordinate: J_{FPE}

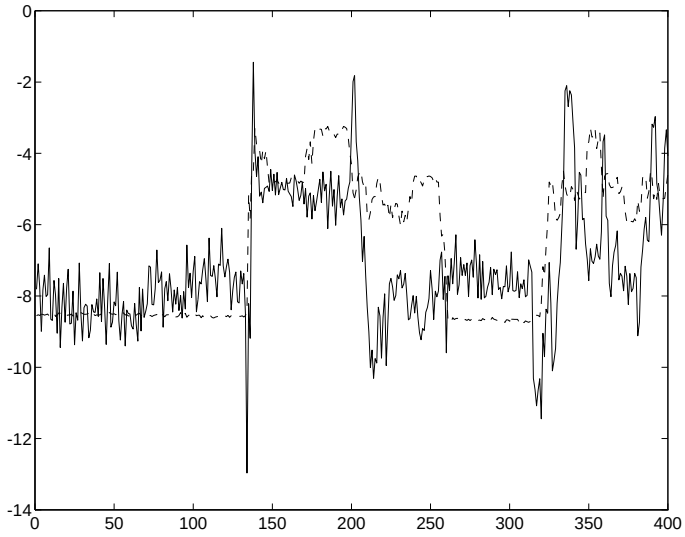

Figure 10: Result with the semi-physical model on the test data set. The solid line represents the true measurement and the dashed line represents the output of the model.

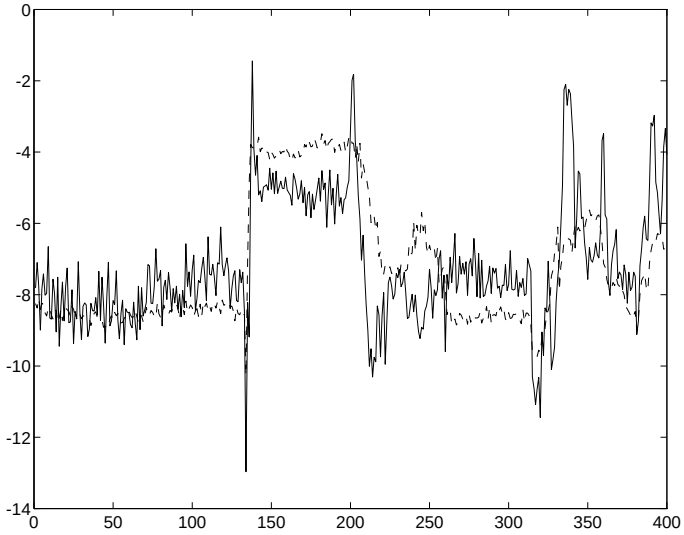

Figure 11: Result with the third degree polynomial model on the test data set. The solid line represents the true measurement and the dashed line represents the output of the model.

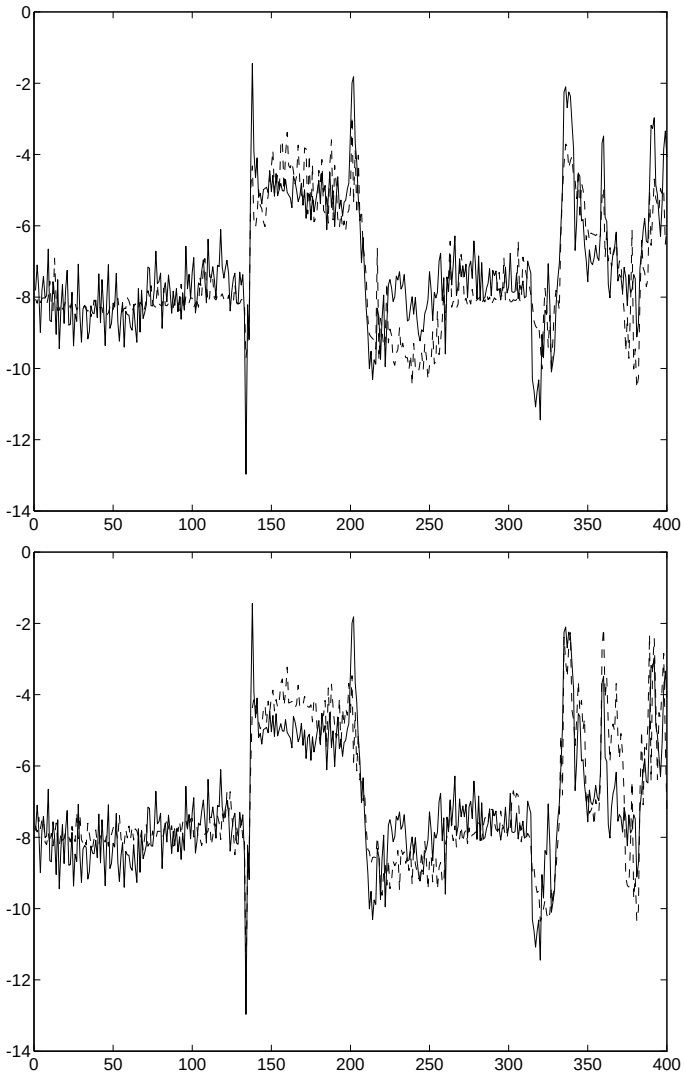

Figure 12: Results with the wavelet network initialized by Algorithm 1 (top) and after 10 iterations of the Gauss-Newton procedure (bottom). The solid lines represent the true measurement and the dashed lines represent the output of the model.

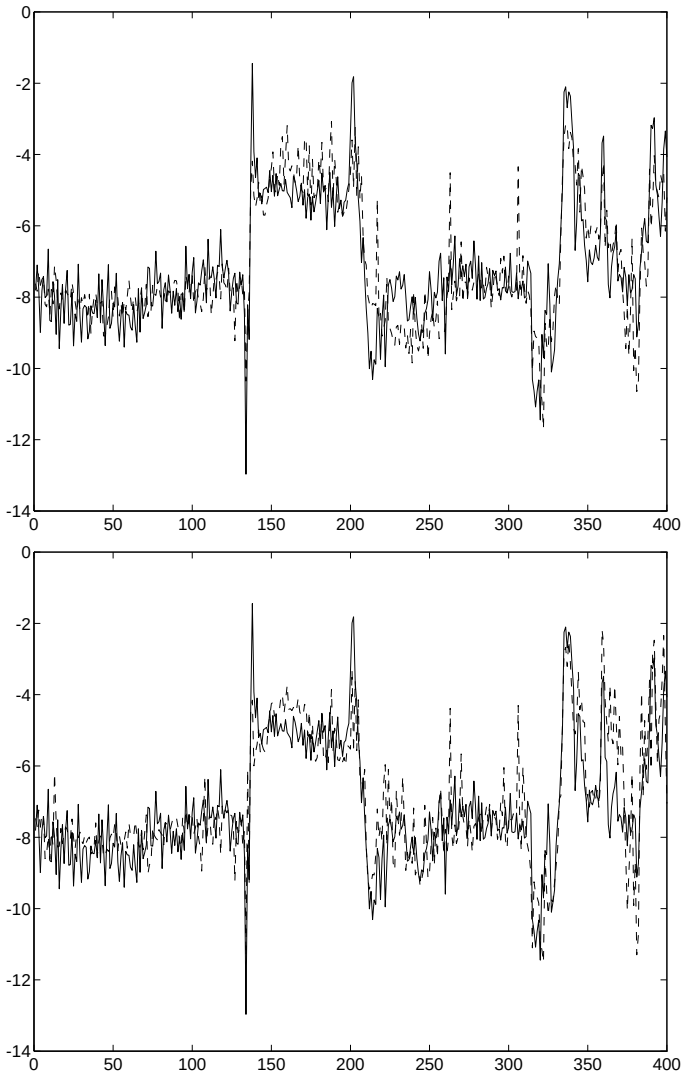

Figure 13: Results with the wavelet network initialized by Algorithm 2 (top) and after 10 iterations of the Gauss-Newton procedure (bottom). The solid lines represent the true measurement and the dashed lines represent the output of the model.

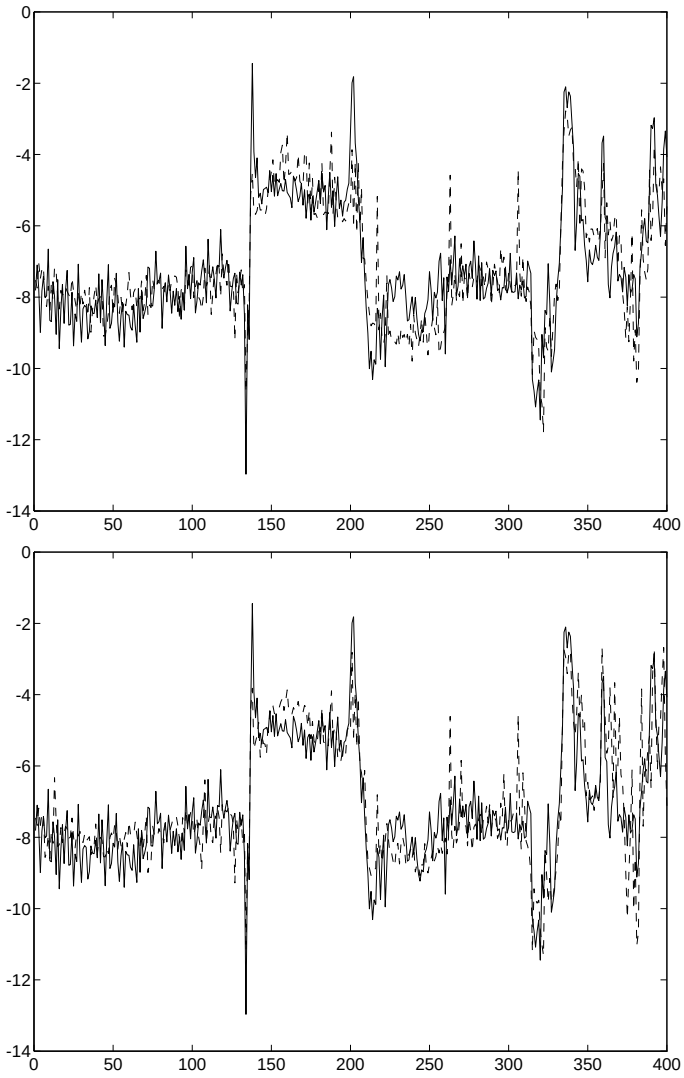

Figure 14: Results with the wavelet network initialized by Algorithm 3 (top) and after 10 iterations of the Gauss-Newton procedure (bottom). The solid lines represent the true measurement and the dashed lines represent the output of the model.

References

[1] C. Chui, *Wavelets : a tutorial in theory and applications.* Inc. Boston, San Diego: Academic Press, 1992.

[2] M. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Raphael, eds., *Wavelets and their applications.* Boston: Jones and Bartlett books in mathematics, 1992.

[3] A. Benveniste, A. Juditsky, B. Delyon, Q. Zhang, and P. Glorennec, "Wavelets in identification," in *SYSID'94, 10th IFAC Symposium on System Identification*, (Copenhagen), 1994.

[4] G. Cybenko, "Approximation by superposition of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, pp. 303–314, 1989.

[5] K. Hornik, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, 1989.

[6] A. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. on Information Theory*, vol. 39, no. 3, 1993.

[7] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481–1497, 1990.

[8] J. Sjöberg, H. Hjalmarsson, and L. Ljung, "Neural networks in identification," in *SYSID'94, 10th IFAC Symposium on System Identification*, (Copenhagen), 1994.

[9] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. on Neural Networks*, vol. 3, pp. 889–898, Nov. 1992.

[10] Q. Zhang, "Wavelet networks : the radial structure and an efficient initialization procedure," Technical Report LiTH-ISY-I-1423, Linköping University, Oct. 1992.

[11] Q. Zhang, "Regressor selection and wavelet network construction," Technical Report 709, Inria, Apr. 1993.

[12] Y. Pati and P. Krishnaprasad, "Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations," *IEEE Trans. on Neural Networks*, vol. 4, pp. 73–85, Jan. 1993.

[13] J. Hong, *Identification of stable systems by wavelet transform and artificial neural networks*. PhD thesis, University of Pittsburg, Ptissburgh, PA, 1992.

[14] B. Bakshi and G. Stephanopoulos, "Wave-net : a multiresolution, hierarchical neural network with localized learning," *American Ins. of Chemical Eng. Journal*, vol. 39, pp. 57–81, Jan. 1993.

[15] I. Daubechies, *Ten lectures on wavelets. CBMS-NSF regional series in applied mathematics*, 1992.

[16] S. Tretter, *Introduction to discrete-time signal processing*. John Wiley & Sons, 1976.

[17] S. Mallat, "Multiresolution approximation and wavelets orthonormal bases of $l^2(r)$," *Trans. Amer. Math. Soc.*, vol. 315, no. 1, pp. 69–8, 1989.

[18] J. Duffin and A. Schaeffer, "A class of nonharmonic fourier series," *Amer. Math. Soc.*, vol. 72, pp. 341–336, 1952.

[19] K. Gröchenig, "Acceleration of the frame algorithm," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, 1993.

[20] B. Delyon, A. Juditsky, and A. Benveniste, "Accuracy analysis for wavelet networks," *IEEE Transactions on neural networks*, 1994. to appear.

[21] A. Juditsky, "Wavelet estimators: adapting to unknown smoothness," Technical Report IRISA, in preparation, 1994.

[22] N. Draper and H. Smith, *Applied regression analysis. Series in Probability and Mathematical Statistics*, Wiley, 1981. Second edition.

[23] J. Friedman and W. Stuetzle, "Projection pursuit regression," *J. Amer. Stat. Assoc.*, vol. 76, pp. 817–823, 1981.

[24] P. Huber, "Projection pursuit (with discussion)," *Ann. Statist.*, vol. 13, pp. 435–475, 1985.

[25] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," Technical Report 619, New-York University, Computer Science Department, Aug. 1993.

[26] S. Qian and D. Chen, "Signal representation using adaptive normalized gaussian functions," *Signal Processing*, vol. 36, March 1994.

[27] S. Chen, S. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[28] S. Chen, C. Cowan, and P. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 2, pp. 302–309, March 1991.

[29] D. M. Young, *Iterative solution of large linear systems*. Academic Press, 1971.

[30] L. Ljung, *System Identification : Theory for the User. Prentice Hall Information and System Sciences Series*, Prentice Hall, 1987.

[31] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[32] H. Akaike, "Fitting autoregressive models for prediction," *Ann. Inst. Stat. Math.*, vol. 21, pp. 243–347, 1969.

[33] Q. Zhang, "Wavenet," Public domain MATLAB toolbox, Anonymous FTP: ftp.irisa.fr : /local/wavenet, 1993.

[34] Q. Zhang, *Contribution à la surveillance de procédés industriels*. PhD thesis, Université de Rennes I, Dec. 1991.

[35] Q. Zhang, M. Basseville, and A. Benveniste, "Early warning of slight changes in systems and plants with application to condition based maintenance," *Automatica*, vol. 30, no. 1, pp. 95–113, 1994. Special Issue on Statistical Methods in Signal Processing and Control.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399