HAL
open science

Learning nonlinear hybrid systems: from sparse optimization to support vector regression

van Luong Le, Fabien Lauer, Laurent Bako, Gérard Bloch

To cite this version:

van Luong Le, Fabien Lauer, Laurent Bako, Gérard Bloch. Learning nonlinear hybrid systems: from sparse optimization to support vector regression. 16th International Conference on Hybrid systems: computation and control, HSCC 2013, Apr 2013, Philadelphia, United States. pp.33-42, 10.1145/2461328.2461336 . hal-00801145

HAL Id: hal-00801145

https://hal.science/hal-00801145v1

Submitted on 15 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Learning Nonlinear Hybrid Systems: from Sparse Optimization to Support Vector Regression

Van Luong Le
Université de Lorraine, CRAN
CNRS
van-luong.le@univ-lorraine.fr

Fabien Lauer
Université de Lorraine, LORIA
CNRS
Inria
fabien.lauer@loria.fr

Laurent Bako
Univ Lille Nord
Mines Douai – URIA
laurent.bako@mines-douai.fr

Gérard Bloch
Université de Lorraine, CRAN
CNRS
gerard.bloch@univ-lorraine.fr

ABSTRACT

This paper deals with the identification of hybrid systems switching between nonlinear subsystems of unknown structure and focuses on the connections with a family of machine learning algorithms known as support vector machines. In particular, we consider a recent approach to nonlinear hybrid system identification based on a convex relaxation of a sparse optimization problem. In this approach, the submodels are iteratively estimated one by one by maximizing the sparsity of the corresponding error vector. We extend this approach in several ways. First, we relax the sparsity condition by introducing robust sparsity, which can be optimized through the minimization of a modified ℓ_1-norm or, equivalently, of the ε-insensitive loss function. Then, we show that, depending on the choice of regularizer, the method is equivalent to different forms of support vector regression. More precisely, the submodels can be estimated by iteratively solving a classical support vector regression problem, in which the sparsity of support vectors relates to the sparsity of the error vector in the considered hybrid system identification framework. This allows us to extend theoretical results as well as efficient optimization algorithms from the field of machine learning to the hybrid system framework.

Categories and Subject Descriptors

G.1 [**Mathematics of Computing**]: Numerical Analysis—*Approximation, Optimization*; I.2.6 [**Artificial Intelligence**]: Learning

This is the author's version of the work. Compared with the published version, this version includes an erratum regarding the equation in section 4.3. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in
HSCC'13, April 8–11, 2013, Philadelphia, Pennsylvania, USA.
Copyright 2013 ACM 978-1-4503-1567-8/13/04 ...$15.00
http://doi.acm.org/10.1145/2461328.2461336.

Keywords

switched nonlinear systems; system identification; switched regression; support vector machine; robustness to noise; sparsity; convex optimization

1. INTRODUCTION

This paper deals with the identification of hybrid dynamical systems and in particular with the connections between this problem and the fields of sparse optimization and machine learning. More precisely, we consider switched nonlinear systems that can be written in input–output Nonlinear ARX (NARX) form as

$$y_i = f_{q_i}(\boldsymbol{x}_i) + v_i, \tag{1}$$

where v_i is a noise term, $q_i \in \{1, \ldots, s\}$ and, at time step i, the output y_i is given by the q_ith function of the collection of submodels $\{f_j\}_{j=1}^s$ and the vector of regressors

$$\boldsymbol{x}_i = [y_{i-1}, \ldots, y_{i-n_a}, u_{i-n_k}, \ldots, u_{i-n_k-n_b+1}]^T$$

built from past inputs u_{i-k} and outputs y_{i-k}. In this setting, we call q_i the mode of $\boldsymbol{x}_i$. The goal of the identification is to estimate the submodels $\{f_j\}_{j=1}^s$ from a training set of input–output data, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, without knowledge of the corresponding sequence of modes $\{q_i\}_{i=1}^N$. The paper further focuses on the case where the submodels are nonlinear functions of arbitrary and unknown structure.

Related work. Many approaches have been proposed over the last decade for the case where the submodels f_j are linear (or affine) [24]. These include the algebraic approach [36] and various convex [22, 1, 23, 18] and nonconvex [9, 26, 4, 13, 17, 15] optimization-based approaches, to name a few. But far fewer works have considered the nonlinear case, since the problem was formally stated more recently in [16] with a preliminary solution suffering from strong limitations, particularly regarding the number of data that could be processed by the identification algorithm. In [19], the approach of [17] for linear hybrid systems is extended to estimate the s nonlinear submodels at once via the solution of a continuous, but nonconvex, optimization problem. The quality of the estimates thus obtained rely on the capabilities of global optimization solvers, and thus cannot be guaranteed

for models with many parameters. On the other hand, the approach of [2], which extends the method of [1], relies on a sequence of convex optimizations to estimate the submodels one by one, and thus does not suffer from local minima issues. More precisely, the method relies on the formulation of the identification as a sparse optimization problem and its convex relaxation. However, the analysis of the method provided in [2] is only valid for noiseless data. This is all the more unsatisfactory in the nonlinear setting, since the uncertainty on the model structure can be interpreted as a form of noise.

Contribution. The present work considers the sparse optimization framework of [1, 2], which we extend in several ways. First, we introduce the notion of robust sparsity to relax the conditions on the noise under which the method can yield optimal estimates. Then, a convex relaxation is proposed to allow for the optimization of the robust sparsity through the minimization of a modified ℓ_1-norm. Finally, we show that the resulting convex optimization programs can be equivalently formulated as the minimization of the ε-insensitive loss function proposed in the machine learning community for Support Vector Regression (SVR) [32], a particular instance of the Support Vector Machine (SVM) [35]. Depending on the choice of regularizer, a formal equivalence between sparse optimization based hybrid system identification and SVR can be obtained, in which the sparsity of support vectors relates to the sparsity of the error vector in the considered identification framework. Algorithmically, the submodels can be estimated by iteratively solving a classical SVR learning problem, which allows the method to benefit from the numerous advances on SVMs for machine learning, for instance regarding the tuning of the hyperparameters. In addition, efficient optimization algorithms dedicated to SVMs can be applied as off-the-shelf solvers to nonlinear hybrid system identification with very large data sets, which typically cannot be handled by general purpose convex optimization solvers.

Notations. All vectors and matrices are written with bold symbols. In particular, $\mathbf{1}$ denotes a vector of appropriate dimension filled with ones and $\boldsymbol{I}$ is the identity matrix. Inequality symbols applied to vectors are to be understood entry-wise. The notation $\langle \cdot, \cdot \rangle_E$ stands for the inner product in E, while $|I|$ denotes the cardinality of I whenever I is a set.

Paper organization. The rest of the paper starts in Sect. 2 with some background in nonlinear model estimation by focusing more particularly on recent machine learning approaches. Then, Section 3 recalls the sparse optimization approach to hybrid system identification and its extension to the case of nonlinear submodels (Sect. 3.2), including a discussion on the choice of the regularizer (Sect. 3.3). Robust sparsity is introduced in Sect. 4, where the connection with SVMs is explicitly stated (Sect. 4.1–4.3) and its benefits discussed (Sect. 4.4 and 4.6). The paper ends with numerical examples in Sect. 5 and conclusions in Sect. 6.

2. LEARNING NONLINEAR MODELS

In this section we provide the necessary background on nonlinear model estimation. In particular, we consider the case of arbitrary and unknown nonlinearities and focus on the estimation of a single (non-hybrid) nonlinear model. In such a setting, both the structure and the parameters of the model must be estimated from the data. While this clearly constitutes a difficulty, recent approaches developed in machine learning allow both subproblems to be solved simultaneously through convex optimization.

One issue which must be considered with care in nonlinear regression compared to the linear case is overfitting. While linear models are constrained within a restricted function class of low capacity, nonlinear function classes can include very complex functions. The typical functions classes provide sufficient flexibility for the model to yield a perfect fit of the data. Thus, if we were to minimize the error on a data set, the model would learn the noise as well as the target function, i.e., overfit the training data. Hence, most approaches to nonlinear modeling include a regularization scheme to control the complexity (or flexibility) of the model.

Formally, the considered approach to nonlinear modeling can be stated as follows. Assume we are given a training set $\mathcal{D}$ of N pairs $(\boldsymbol{x}_i, y_i) \in (\mathcal{X} \subset \mathbb{R}^d) \times (\mathcal{Y} \subset \mathbb{R})$, $i = 1, \ldots, N$, with the general goal of learning a function $f \in \mathcal{H}$ such that this function minimizes, over some function class $\mathcal{H}$, a regularized functional representing a trade-off between the fit to the data and some regularity conditions of f:

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i), y_i) + \lambda \mathcal{R}(f), \tag{2}$$

where the data term is defined through a loss function ℓ of $\mathbb{R}^2$ to $\mathbb{R}^+$, $\mathcal{R}(f)$ is a general regularization term and $\lambda \geq 0$ tunes the trade-off between the two terms.

Though searching for f within a specific function class $\mathcal{H}$ can be related in some cases to a particular choice of structure for the nonlinear model f, this can also be more general. In particular, by assuming that f is an expansion over some functional basis, a single function $f \in \mathcal{H}$ can have multiple representations (and parametrizations) depending on the choice of the basis. In addition, we will see below that $\mathcal{H}$ can be an infinite dimensional function space with the universal approximation capacity while still allowing for learning from a finite set of data. As a practical consequence, arbitrary nonlinearities can be learned without introducing a bias due to an arbitrary choice of unsuitable or insufficiently flexible structure for f.[1]

2.1 Learning in RKHS

We start with some formal definitions. Let K be a real-valued positive type (or positive definite) function [5] on $\mathcal{X}^2$ and $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ the corresponding reproducing kernel Hilbert space (RKHS), i.e., K is the reproducing kernel of $\mathcal{H}$ with the reproducing property: $\forall \boldsymbol{x} \in \mathcal{X}$, $\forall f \in \mathcal{H}$, $\langle f, K(\boldsymbol{x}, \cdot) \rangle_{\mathcal{H}} = f(\boldsymbol{x})$, and in particular

$$\langle K(\boldsymbol{x}, \cdot), K(\boldsymbol{x}', \cdot) \rangle_{\mathcal{H}} = K(\boldsymbol{x}, \boldsymbol{x}').$$

In this case, the class of functions $\mathcal{H}$ can be written as

$$\mathcal{H} = \left\{ f : f = \sum_{i=1}^{\infty} \alpha_i K(\boldsymbol{x}_i, \cdot),\ \alpha_i \in \mathbb{R}, \boldsymbol{x}_i \in \mathcal{X}, \|f\|_{\mathcal{H}} < +\infty \right\}, \tag{3}$$

where the norm $\|\cdot\|_{\mathcal{H}}$ is the norm in $\mathcal{H}$ induced by the inner

[1]Note that, due to the bias-variance dilemma, this does not imply the optimal recovery of the target function.

product defined as

$$\|f\|_{\mathcal{H}}^2 = \langle f, f\rangle_{\mathcal{H}} = \sum_{i=1}^{\infty}\sum_{j=1}^{\infty} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{4}$$

Typical examples of kernel functions include the Gaussian Radial Basis Function (Gaussian RBF) kernel, $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2/2\sigma^2)$, the polynomial kernel, $K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T\boldsymbol{x}'+1)^\gamma$, and the linear kernel, $K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T\boldsymbol{x}'$. For the Gaussian RBF kernel, the space $\mathcal{H}$ consists of all infinitely differentiable functions of $\mathcal{X}$ and thus enjoys the so-called universal approximation capacity, i.e., an arbitrary function can be arbitrarily well approximated by a function in $\mathcal{H}$.

When learning in an RKHS, a natural choice for $\mathcal{R}(f)$ is based on the RKHS norm:

$$\mathcal{R}(f) = \frac{1}{2}\|f\|_{\mathcal{H}}^2. \tag{5}$$

Such a regularizer is a measure of the function smoothness[2] and is particularly suitable for cases without prior information on the shape of the target function. In addition, with (5), the representer theorem [27] provides an explicit structure for the solution to (2). This theorem is recalled below, where $\mathcal{D}_x$ denotes the set of all points $\boldsymbol{x}_i$ in the training set $\mathcal{D}$ (a sketch of the proof is given in Appendix A).

THEOREM 2.1 (REPRESENTER THEOREM, [27]). *The solution f^* to (2), with $\mathcal{H}$ defined as in (3), $\mathcal{R}(f) = g(\|f\|_{\mathcal{H}})$ and a monotonically increasing function $g : \mathbb{R}^+ \to \mathbb{R}^+$, is a kernel expansion over the training set, i.e., f^* is in the span of $\{K(\boldsymbol{x}_i, \cdot) \ : \ \boldsymbol{x}_i \in \mathcal{D}_x\}$.*

This result shows that minimizing any regularized functional of the form (2) over an RKHS leads to finite linear combinations of kernel functions computed at the training points:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}). \tag{6}$$

Note that a semiparametric version of Theorem 2.1 is also provided in [27] to allow for a bias term in the model. This is obtained by considering a model $\tilde{f} = f + b$, with $f \in \mathcal{H}$ and $b \in \mathbb{R}$, regularized only in f. In most of the paper, we omit this straightforward substitution and focus on models in the form of f in (6).

Finally, given the structure of the model (6), solving (2) with a convex loss function ℓ amounts to a finite-dimensional and convex optimization problem.

2.2 ℓ_1-norm regularization

Another typical regularization scheme for models based on kernel functions is to penalize the ℓ_1-norm of the parameters, i.e., to penalize $\|\boldsymbol{\alpha}\|_1$, with $\boldsymbol{\alpha} = [\alpha_1 \ \dots \ \alpha_N]^T$, in (6). However, this scheme cannot apply to (2) with $\mathcal{H}$ defined as in (3), since $\boldsymbol{\alpha}$ depends on the particular choice of basis functions through $\{\boldsymbol{x}_i\}_{i=1}^{\infty}$ and need not be uniquely defined. Therefore, this scheme is usually applied with the structure of f fixed a priori. With f chosen as in (6), this leads to

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \ \sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i), y_i) + \lambda\|\boldsymbol{\alpha}\|_1 \tag{7}$$
$$\text{s.t.} \quad f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}).$$

While this learning strategy also provides control over the complexity of the model, as detailed in Appendix B, it is often chosen in order to favor sparse solutions with few nonzero α_i. Indeed, the sparsity of $\boldsymbol{\alpha}$ directly defines the number of operations required to compute the output of the model (6) for a given $\boldsymbol{x}$.

3. SPARSE OPTIMIZATION FOR HYBRID SYSTEM IDENTIFICATION

We now recall the sparse optimization framework of [1] in the case of switched linear systems, while its extension to the case of switched nonlinear systems will be detailed in Sect. 3.2.

Consider a switched linear system, i.e., of the form (1) with linear submodels, $f_j(\boldsymbol{x}_i) = \boldsymbol{x}_i^T\boldsymbol{\theta}_j$. As proposed in [1], switched linear systems can be identified via sparse optimization. In this approach, a single parameter vector $\boldsymbol{\theta}$ is first estimated by maximizing the sparsity of the error vector, $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}$, where $\boldsymbol{X} = [\boldsymbol{x}_1 \ \dots \ \boldsymbol{x}_N]^T$ and $\boldsymbol{y} = [y_1 \ \dots \ y_N]^T$. In the noiseless case, each entry e_i in $\boldsymbol{e}$ can be zero by choosing $\boldsymbol{\theta}$ as the vector of parameters $\boldsymbol{\theta}_{q_i}$ that generated the corresponding data point. Therefore, by searching for the pair $(\boldsymbol{\theta}, \boldsymbol{e})$ leading to the sparsest vector $\boldsymbol{e}$, we recover the parameters of the submodel that generated the largest percentage of data. Formally, the sparsity is measured through the ℓ_0-pseudo norm,

$$\|\boldsymbol{e}\|_0 = |\{i : e_i \neq 0\}|,$$

where a vector $\boldsymbol{e}$ with small norm $\|\boldsymbol{e}\|_0$ is said to be sparse, and one solves

$$\min_{\boldsymbol{\theta},\boldsymbol{e}} \ \|\boldsymbol{e}\|_0 \tag{8}$$
$$\text{s.t.} \quad \boldsymbol{e} = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}$$

to obtain the first parameter vector. Then, the data points with corresponding $e_i = 0$ are removed from the data set, and (8) is solved again to obtain the second parameter vector. Applying this procedure iteratively until all data are correctly approximated and removed from the training set yields all the submodels.

3.1 Convex relaxation

Since (8) is a nonconvex optimization problem, we instead consider a convex relaxation based on the best convex approximation to the ℓ_0-pseudo norm, i.e., the ℓ_1-norm. In order to improve the sparsity of the solution, an iteratively reweighted scheme is employed, as proposed by [7]. Thus, each parameter vector is recovered by iteratively solving

$$\min_{\boldsymbol{\theta}} \ \|\boldsymbol{W}^k(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta})\|_1, \tag{9}$$

where $\boldsymbol{W}^0 = \boldsymbol{I}$ and $\boldsymbol{W}^k$ is a diagonal matrix of entries $(W^k)_{ii} = 1/(|y_i - \boldsymbol{x}_i^T\boldsymbol{\theta}^{k-1}| + \delta)$, with δ a small positive number, and $\boldsymbol{\theta}^{k-1}$ the solution at iteration $k-1$.

In [1], the following sparse recovery conditions are stated.

[2]For smooth kernel functions K, all $f \in \mathcal{H}$ are smooth functions with infinite order of differentiability. However, here, a large measure of smoothness refers to functions with derivatives of small magnitude rather than a high order of differentiability. The RKHS norm in (5) provides an upper bound on these magnitudes (see, e.g., Corollary 4.36 in [33]).

THEOREM 3.1 (THEOREM 11 IN [1]). *If there is a vector $\boldsymbol{\theta}$ such that*

$$\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|_0 < \frac{1}{2}\left(1 + \frac{1}{m(\boldsymbol{X})}\right),$$

with

$$m(\boldsymbol{X}) = \max_{1\le t,k\le N, t\neq k} \frac{|M_{tk}|}{\sqrt{(1-M_{tt})(1-M_{kk})}},$$

where $\boldsymbol{M} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ and M_{tk} is the (t,k)th entry of $\boldsymbol{M}$, then $\boldsymbol{\theta}$ is the unique solution to both (8) *and* (9) *with $\boldsymbol{W}^k = \boldsymbol{I}$.*

Note that this result directly applies only to the first iteration of the reweighted scheme. However, it provides sufficient ground for the method, while the convergence analysis of the reweighted scheme remains a difficult open issue.

3.2 Extension to nonlinear submodels

When the submodels f_j are nonlinear, the procedure above can be extended to estimate nonlinear submodels. The basic idea is to replace $\boldsymbol{x}_i^T\boldsymbol{\theta}$ by an expansion over a set of basis functions, e.g., a kernel expansion as in (6). As discussed in Sect. 2, depending on the regularizer $\mathcal{R}(f)$, this either corresponds to an arbitrary choice of nonlinear structure for the model or to the explicit form of the solution. We first describe the complete procedure for a general regularizer and nonlinear model before detailing the typical choices.

For a given function class $\mathcal{H}$, the nonlinear submodel f of a single mode is estimated by solving

$$\min_{f\in\mathcal{H}} \sum_{i=1}^{N} w_i|y_i - f(\boldsymbol{x}_i)| + \lambda\mathcal{R}(f), \tag{10}$$

with $w_i = (W^k)_{ii} > 0$. By defining the error vector $\boldsymbol{e} \in \mathbb{R}^N$ with components $e_i = y_i - f(\boldsymbol{x}_i)$, we see that the first term in (10) is merely $\|\boldsymbol{W}^k\boldsymbol{e}\|_1$.

3.3 Choice of the regularizer

In machine learning, it is a well-known fact that one cannot learn without a minimal set of assumptions on the target function. In the most general case, where no prior knowledge is available, the less informative assumption concerns the smoothness of the target function. Indeed, without assuming that function values should be close for two points that are close in the regression space $\mathcal{X}$, one cannot learn from a finite set of points and generalize to others. In practice, the smoothness assumption is typically implemented by regularization as in (2). We now discuss two particular choices for the regularizer $\mathcal{R}(f)$.

3.3.1 Sparsity inducing regularization

In [2], a regularization term based on the ℓ_0-norm of the parameter vector $\boldsymbol{\alpha}$ is introduced, before being relaxed to the convex ℓ_1-norm. While the ℓ_1-norm is a typical choice for regularization, which is also known for its sparsity inducing feature, ℓ_0-norm regularization is more ambiguous regarding the resulting smoothness of f. Therefore, in this case, the aim of minimizing the ℓ_1-norm is not to recover the smallest ℓ_0-norm solution through a convex relaxation, and we will not delve into theoretical guarantees of convergence of the ℓ_1-solution to the ℓ_0-solution.

Let $\boldsymbol{K}$ be the so-called Gram matrix of the kernel K with respect to the sample $\mathcal{D}_x$, i.e., with all components given by $(\boldsymbol{K})_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Then, with ℓ_1-norm regularization, the submodels are estimated by solving

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \|\boldsymbol{W}^k(\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha})\|_1 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{11}$$

where the classical reweighting scheme applies.

3.3.2 Capacity control regularization

The typical approach used in machine learning to estimate nonlinear functions is to control the capacity of the model by penalizing the nonsmoothness of f. This can be measured through a norm of f.

Using the natural RKHS squared norm defined in (4), $\mathcal{R}(f) = \frac{1}{2}\|f\|_{\mathcal{H}}^2$, the nonlinear submodels are estimated by solving the convex optimization problem

$$\min_{f\in\mathcal{H}} \sum_{i=1}^{N} w_i|y_i - f(\boldsymbol{x}_i)| + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2, \tag{12}$$

whose solution is in the form of (6) by application of Theorem 2.1 with the convex loss function[3] $\ell(f(\boldsymbol{x}_i), y_i) = |y_i - f(\boldsymbol{x}_i)|$ in (2).

4. ROBUST SPARSITY

A preliminary condition to the sparse recovery conditions derived in Sect. 3.1 (see Theorem 3.1) is that the data must be noiseless. Indeed, with noisy data, no (or very few) entries of the error vector can be zero[4], hence breaking the sparsity of the optimal solution.

In order to circumvent the issue of the lack of zeros in the error vector, we introduce robust sparsity as defined through the pseudo-norm

$$\|\boldsymbol{e}\|_{0,\varepsilon} = |\{i \ : \ |e_i| > \varepsilon\}|.$$

Under a bounded noise assumption of the type $\|\boldsymbol{v}\|_\infty = \max_{i\in\{1,\dots,N\}}|v_i| \le \varepsilon$, the error vector, $\boldsymbol{e} = [y_1 - f(\boldsymbol{x}_1), \ \dots, \ y_N - f(\boldsymbol{x}_N)]^T$, can be *robustly sparse*, i.e., with a small value of $\|\boldsymbol{e}\|_{0,\varepsilon}$, if f is a sufficiently good approximation of one of the target submodels f_j.

Instead of the nonconvex pseudo-norm above, we consider the following convex relaxation based on a modified ℓ_1-norm:

$$\|\boldsymbol{e}\|_{1,\varepsilon} = \sum_i (|e_i| - \varepsilon)_+ = \sum_i \max\{0, \ |e_i| - \varepsilon\},$$

which is defined as a sum of pointwise maximum of convex functions of e_i and hence is convex with respect to all components e_i. In the following, we will refer to the pseudo norm above as the $\ell_{1,\varepsilon}$-norm.

With these definitions, the procedure to estimate the submodels under noisy conditions is similar to the one presented in Sect. 3 for the noiseless case, with the $\ell_{1,\varepsilon}$-norm substituted for the ℓ_1-norm. Similarly, after the estimation of a submodel f, the data points correctly approximated by f are removed, where "correctly approximated" is now implemented by the test $|y_i - f(\boldsymbol{x}_i)| \le \varepsilon$.

[3]For the sake of brevity, we omitted the weights w_i in (2), but Theorem 2.1 in its original version found in [27] equivalently applies to a weighted sum of losses.

[4]With nonlinear models of sufficient capacity, the error vector can actually be zero. But, as already discussed, this is not a desirable case, since this would clearly indicate overfitting. Here, we focus on sufficiently regularized (and desirable) solutions, for which the error vector cannot be sparse.

4.1 ℓ_1-norm regularization

By assuming submodels in the form of (6), estimating one of the nonlinear submodels by maximizing the robust sparsity of the error vector can be set as the convex optimization problem:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \ \|\boldsymbol{y}-\boldsymbol{K}\boldsymbol{\alpha}\|_{1,\varepsilon}+\lambda\|\boldsymbol{\alpha}\|_1, \tag{13}$$

where the convexity of the first term is due to the convexity of the $\ell_{1,\varepsilon}$-norm and the linearity of f wrt. the parameters $\boldsymbol{\alpha}$. Note that robust sparsity is only considered for the error vector, and that the standard ℓ_1-norm is used for regularization.

Connection with Support Vector Machines.

Problem (13) can be written as the linear program

$$\begin{aligned}\min_{(\boldsymbol{\alpha},\boldsymbol{a},\boldsymbol{\xi})\in\mathbb{R}^{3N}} \quad & \mathbf{1}^T\boldsymbol{a}+C\mathbf{1}^T\boldsymbol{\xi} \\ \text{s.t.} \quad & -\boldsymbol{\xi}-\varepsilon\mathbf{1}\leq\boldsymbol{y}-\boldsymbol{K}\boldsymbol{\alpha}\leq\varepsilon\mathbf{1}+\boldsymbol{\xi} \\ & -\boldsymbol{a}\leq\boldsymbol{\alpha}\leq\boldsymbol{a},\end{aligned} \tag{14}$$

with $C=1/\lambda$. Here, the objective function has been divided by λ in order to emphasize the equivalence with the training algorithm of the so-called Linear Programming Support Vector Regression (LP-SVR) proposed in [21] for nonlinear function approximation.

4.2 RKHS norm regularization

Introducing robust sparsity in (12) yields

$$\min_{f\in\mathcal{H}} \ \frac{1}{2}\|f\|^2_{\mathcal{H}}+C\sum_{i=1}^N \ell_\varepsilon(y_i,f(\boldsymbol{x}_i)), \tag{15}$$

with $C=1/\lambda$ and the ε-insensitive loss function defined as in [35] by

$$\ell_\varepsilon(y_i,f(\boldsymbol{x}_i))=\max\left\{0,\ |y_i-f(\boldsymbol{x}_i)|-\varepsilon\right\}.$$

With these definitions, the connection with Support Vector Regression (SVR) [32] becomes apparent, as detailed below.

Connection with Support Vector Machines and explicit solution.

It is known (see, e.g., [32]) that a kernel function K implicitly defines a (nonlinear) feature map, $\Phi:\mathcal{X}\to\mathcal{F}$, mapping the regressors $\boldsymbol{x}$ into a feature space $\mathcal{F}$, where the model f becomes linear, i.e.,

$$f(\boldsymbol{x})=\langle\boldsymbol{w},\Phi(\boldsymbol{x})\rangle_{\mathcal{F}}, \tag{16}$$

with parameters $\boldsymbol{w}\in\mathcal{F}$. In order to emphasize the relationship with SVMs below, we further consider the affine model $\tilde{f}=f+b$, with $b\in\mathbb{R}$. With these notations, and a simple substitution of $\tilde{f}$ for f in the computation of the loss, problem (15) can be written as

$$\begin{aligned}\min_{\boldsymbol{w},b,\xi_i,\xi_i'} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2+C\sum_{i=1}^N(\xi_i+\xi_i') \\ \text{s.t.} \quad & y_i-\langle\boldsymbol{w},\Phi(\boldsymbol{x}_i)\rangle_{\mathcal{F}}-b\leq\varepsilon+\xi_i \\ & y_i-\langle\boldsymbol{w},\Phi(\boldsymbol{x}_i)\rangle_{\mathcal{F}}-b\geq-\varepsilon-\xi_i' \\ & \xi_i\geq 0,\ \xi_i'\geq 0,\end{aligned} \tag{17}$$

which is the primal form of the training algorithm of a support vector machine for nonlinear regression (SVR) [32]. Note that, Φ and $\mathcal{F}$ are only implicit and need not be known nor finite-dimensional, and so does $\boldsymbol{w}$. What is known however is that, by construction, $\langle\Phi(\boldsymbol{x}),\Phi(\boldsymbol{x}')\rangle_{\mathcal{F}}=K(\boldsymbol{x},\boldsymbol{x}')$. Thus, by Lagrangian duality, this problem can be reformulated as the finite-dimensional quadratic program

$$\begin{aligned}\max_{\boldsymbol{\beta}\in\mathbb{R}^N,\boldsymbol{\beta}'\in\mathbb{R}^N} \quad & -\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}')^T\boldsymbol{K}(\boldsymbol{\beta}-\boldsymbol{\beta}')-\varepsilon\mathbf{1}^T(\boldsymbol{\beta}+\boldsymbol{\beta}') \\ & +\boldsymbol{y}^T(\boldsymbol{\beta}-\boldsymbol{\beta}') \\ \text{s.t.} \quad & \mathbf{1}^T(\boldsymbol{\beta}-\boldsymbol{\beta}')=0 \\ & 0\leq\boldsymbol{\beta}\leq C,\ 0\leq\boldsymbol{\beta}'\leq C,\end{aligned} \tag{18}$$

which involves Φ only through the (computable) matrix $\boldsymbol{K}$. Then, the solution of the primal is given by $\boldsymbol{w}=\sum_{i=1}^N\alpha_i\Phi(\boldsymbol{x}_i)$, where $\alpha_i=\beta_i-\beta_i'$. The reader is referred to [32] for more details on the derivation of (18) and on the computation of b.

From these results, the connection with RKHS theory (Sect. 2.1) can readily be seen by choosing Φ as the most natural feature map for a kernel function, i.e., the one that maps $\mathcal{X}$ to the corresponding RKHS: $\Phi(\boldsymbol{x})=K(\boldsymbol{x},\cdot)$ and $\mathcal{F}=\mathcal{H}$. Indeed, this yields $f=\boldsymbol{w}$ and $f(\boldsymbol{x})=\langle f,K(\boldsymbol{x},\cdot)\rangle_{\mathcal{H}}=\langle\boldsymbol{w},\Phi(\boldsymbol{x})\rangle_{\mathcal{F}}=\sum_{i=1}^N\alpha_i\langle\Phi(\boldsymbol{x}_i),\Phi(\boldsymbol{x})\rangle_{\mathcal{F}}=\sum_{i=1}^N\alpha_iK(\boldsymbol{x}_i,\boldsymbol{x})$.

4.3 Sparsity, support vectors and outliers

We now detail the connections between nonlinear hybrid system identification and support vector regression at the sparsity level.

In the robust sparsity optimization framework, the error vector, $\boldsymbol{e}$, is (robustly) sparse and the data points with large errors, i.e., $|y_i-f(\boldsymbol{x}_i)|>\varepsilon$, are considered as outliers. In the SVR framework, these points are known as *support vectors* (SVs) and the model f is said to be sparse in the sense that the vector of parameters $\boldsymbol{\alpha}$ is sparse. Formally, a SV is a regression vector $\boldsymbol{x}_i$ from the training set which is retained in the model f after training, i.e., the set of SVs is the set of points $\boldsymbol{x}_i$ for which $\alpha_i\neq 0$. Sparsity of the model is an advantageous feature of SVR which leads to faster computations of outputs $f(\boldsymbol{x})$ by reducing the number of terms in the sum (6). The connection between the two forms of sparsity, measured by $\|\boldsymbol{\alpha}\|_0$ for the model and by $\|\boldsymbol{e}\|_{0,\varepsilon}$ for the error, is given by the following classical result (see, e.g., [32]):

$$\|\boldsymbol{\alpha}\|_0=\|\boldsymbol{e}\|_{0,\varepsilon}+(N-\|\boldsymbol{e}-\varepsilon\mathbf{1}\|_0)+(N-\|\boldsymbol{e}+\varepsilon\mathbf{1}\|_0).$$

In words, the set of SVs coincides with the set of points that are not inside the ε-tube of insensitivity, i.e., points with $|y_i-f(\boldsymbol{x}_i)|\geq\varepsilon$. This set differs from the set of outliers only by the points that lie exactly on the boundary of the ε-tube.

4.4 Tuning of the threshold ε

Regarding the tuning of the threshold ε, the following different cases must be considered.

4.4.1 Bounded-error approach

In [4], a bounded-error approach is proposed for linear hybrid system identification, in which the number of submodels is estimated in order to satisfy, for a predefined threshold δ, a bound of the form $|y_i-f(\boldsymbol{x}_i)|<\delta$, for all data points. Such a bound is optimal in the bounded noise case, with $\delta=\|\boldsymbol{v}\|_\infty$, where $\boldsymbol{v}$ is the concatenation of all noise terms. But it is also more general in the sense that it does not require a noise model. Indeed, the aim is to obtain a set of

submodels which approximate the data with a given tolerance. Thus, the parameter δ allows one to tune the trade-off between the model complexity (measured as the number of submodels) and the fit to the data. The original sparse optimization approach of [1] is of a similar flavor, but uses a data-dependent threshold $\delta(\boldsymbol{x}_i, y_i, f)$. In addition, its analysis focuses on the noiseless case, in which the true number of modes can be recovered.

Following these works, a similar strategy applies to the proposed method, where ε plays a similar role as δ.

4.4.2 With assumptions on the noise model

Optimal values for ε have been investigated in the context of SVR under various noise models by different authors, where "optimal" is to be understood with respect to the precise setting of each author's analysis. A common result of these works shows a linear dependency between the optimal value of ε and the noise standard deviation, σ_v. While this is rather intuitive for a uniform noise model, in which case the optimum is $\varepsilon = \sigma_v$, this is more intricate for Gaussian noise. In particular, [30] obtained a first estimate at $\varepsilon = 0.621\sigma_v$, in a maximum likelihood estimation framework. More precisely, they considered the asymptotically optimal ε as the maximizer of the statistical efficiency of the estimator of a single location parameter, which is an oversimplification of the regression setting. A better estimate of the optimal value of ε for Gaussian noise (also in better accordance with experimental observations) was obtained in [14] by considering the complete regression problem in the maximum a posteriori setting. In this case, they estimated the optimal value of ε with a type 2 maximum likelihood method from Bayesian statistics. Another intuitive result developed in [14] concerns Laplacian noise, for which $\varepsilon = 0$ is the optimal choice. Indeed, with $\varepsilon = 0$, the $\ell_{1,\varepsilon}$-norm coincides with the ℓ_1-norm, and the ε-insensitive loss with the absolute loss which is known to yield a maximum likelihood estimator for Laplacian noise in linear regression. These results are summarized in Table 1.

Table 1: Optimal values of ε with a noise model of standard deviation equal to σ_v.

Noise model	Uniform	Laplacian	Gaussian
Optimal ε	σ_v	0	$1.0043\sigma_v$

An additional difficulty with Gaussian or Laplacian noise is that the criterion, $|y_i - f(\boldsymbol{x}_i)| \leq \varepsilon$, used to remove data points after the estimation of a submodel becomes suboptimal: data points with larger noise terms are not removed. In this case, the complete procedure must be stopped when a sufficiently small, but not too small, number of data points remain in the data set. The rationale here is that points with a low noise magnitude are used to estimate the submodels, while the others are considered as outliers. We further assume that these outliers represent a small fraction of the data set. Since at each iteration of the sparse optimization approach, a submodel is estimated from a set of points representing a large fraction of the remaining data, it is expected that outliers are left unused until the end of the procedure.

4.4.3 Automatic tuning

In the SVM literature, problem (17) is sometimes referred to as ε-SVR to distinguish it from the alternative ν-SVR [28] which allows for the automatic tuning of ε. In the derivation of ν-SVR, the trick is to add a term in the objective function of (17) in order to minimize ε while learning the model. This leads to

$$\min_{f\in\mathcal{H},\varepsilon\in\mathbb{R}^+} \frac{1}{2}\|f\|_{\mathcal{H}}^2 + \frac{C}{N}\sum_{i=1}^{N} \ell_\varepsilon(y_i, f(\boldsymbol{x}_i)) + C\nu\varepsilon, \qquad (19)$$

where $\nu \geq 0$ is a new hyperparameter tuning the trade-off between the minimization of ε and the minimization of the errors larger than ε. As for the ε-SVR, the solution to this new fomulation is obtained in the form of (6) by solving the dual. However, in this case, the hyperparameter ν enjoys a number of properties which can ease its tuning when compared to ε in (17). In particular, it is shown in [28] that $\nu > 1$ yields $\varepsilon = 0$ and that, if $\varepsilon > 0$, $\nu \in [0, 1]$ can be interpreted as the fraction of data points outside of the ε-tube of insensitivity, i.e., $\nu \approx \|\boldsymbol{e}\|_{0,\varepsilon}$.

A similar approach can be followed in the case of ℓ_1-norm regularization. This leads to a formulation of the LP-SVR allowing for the automatic tuning of ε via linear programming as proposed in [31] or [21].

4.5 Iteratively reweighted scheme

As in the classical case for sparse optimization, a reweighted scheme can be used to improve the recovery of robustly sparse solutions with low sparsity. This leads for instance to

$$\min_{f\in\mathcal{H}} \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C\sum_{i=1}^{N} w_i\ \ell_\varepsilon(y_i, f(\boldsymbol{x}_i)),$$

with w_i defined as in (10). Such a formulation corresponds to a Weighted-SVR, which has been proposed (with fixed weights) by [34] and others in order to deal with a varying confidence in the data points or to introduce various forms of prior knowledge.

4.6 Algorithmic and implementation issues

The theoretical equivalence between nonlinear hybrid system identification via sparse optimization and support vector regression also yields direct algorithmic benefits. In particular, this means that the problem can be solved efficiently even for large data sets (e.g., with more than ten thousand points).

First, note that all the considered convex formulations, e.g., (11) or (13), are theoretically simple optimization problems due to their convexity. However, despite the possibility to write them as linear programs, e.g., as in (14), solving large-scale instances of such problems requires much more care in practice. In particular, a major issue concerns the memory requirements: the data of the problem, including the (typically dense) N-by-N Gram matrix $\boldsymbol{K}$, simply cannot be stored in the memory of most computers. This basic limitation prevents any subsequent call to a general purpose optimization solver in many cases.

On the other hand, dedicated optimization algorithms have been proposed to train SVMs and benefit from numerous advances in this active field of research, see, e.g., [6]. SVM algorithms typically use decomposition techniques such as sequential minimal optimization (SMO) [25, 29] to avoid the storage of the matrix $\boldsymbol{K}$ in memory. With a proper working set selection strategy, the solution can even be found without having to compute all the elements of the matrix

$\boldsymbol{K}$, thus reducing both the memory and computing load. Good SVM solvers implementing these ideas are for instance SVM^{light} [12] or LibSVM [8]. The latter also implements the Weighted-SVR and can be used in the iteratively reweighted version of the procedure for hybrid system identification, as discussed in Sect. 4.5. For ℓ_1-norm regularization, efficient algorithms are developed in [21]. Finally, these solvers also apply to the original sparse optimization approach of [2] (without robust sparsity) simply by setting $\varepsilon = 0$.

Thus, by showing the equivalence between the robust sparsity optimization approach and support vector regression, we also make the problem tractable for off-the-shelf (and usually freely available) solvers.

5. NUMERICAL SIMULATIONS

We now turn to illustrative examples of application with static functions (Sect. 5.1) and with switching dynamical systems (Sect. 5.2).

5.1 Static example

The first illustrative example considers the approximation of two overlapping nonlinear functions (a sinusoid and a quadratic) from a set of 3000 data points with Gaussian noise of standard deviation $\sigma_v = 0.5$. The submodels are estimated by solving (18) with LibSVM for a Gaussian RBF kernel ($\sigma = 0.5$), $C = 100$ and ε set as in Table 1 for the Gaussian noise model ($\varepsilon = 0.50215$). The first row of Figure 1 shows the first submodel obtained when either one of the two functions dominates the other in terms of the fraction of data points. In both cases, the method correctly estimates the submodel corresponding to the dominating mode. Then, after removing the points close to this submodel, a second submodel is estimated. By thresholding the absolute error, $|y_i - f(\boldsymbol{x}_i)|$, at either 3ε (2nd row of Fig. 1) or ε (last row) in the test for removing points, a sufficient fraction of data are eliminated to allow for the recovery of the second submodel. However, with a threshold of ε, a significant fraction (a bit less than 1/3) of the data remains at the end of the procedure. Then, either the number of submodels is assumed fixed to 2 and the algorithm returns the 2 submodels, or the bounded-error approach is applied and the algorithm continues to estimate additional submodels until all the data are removed.

In this example, we observed that the reweighted scheme of Sect. 4.5 slightly improves the submodels, while the first iteration already yields a satisfactory discrimination between the two modes due to the large fraction of points associated to the dominating one (about 66%). Figure 2 shows the influence of reweighting when this fraction is closer to 50%. For 50.25%, the first iteration is not very accurate, but 10 iterations of the reweighted scheme provide a good approximation of the target submodel. For exactly 50% of data of each mode, the estimated model switches between the two target submodels and cannot discriminate between the modes.

5.2 Switched nonlinear system examples

We now consider the switched nonlinear system example from [19], where the aim is to identify a dynamical system

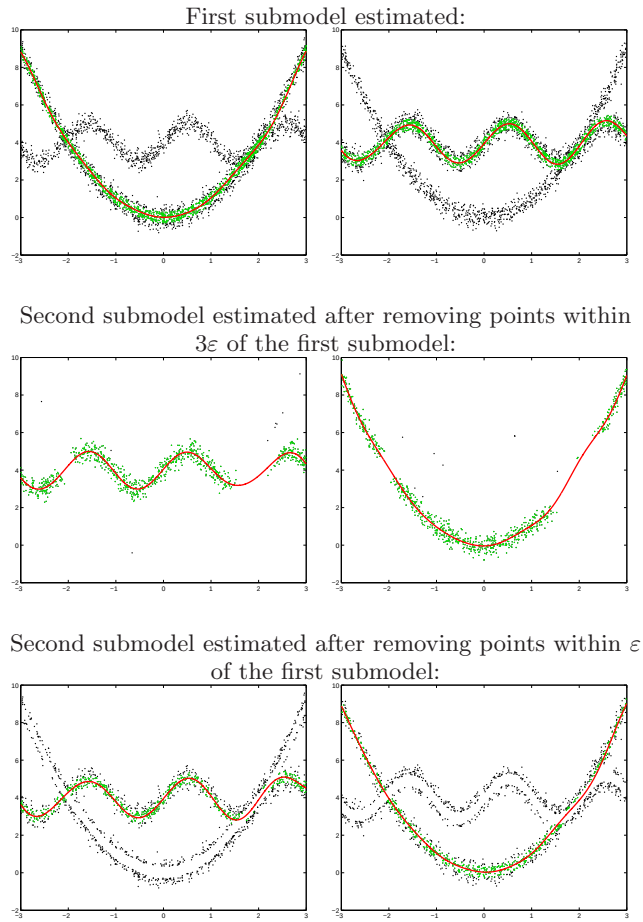

Figure 1: Illustration of the procedure depending on which one of the quadratic (left column) or the sinusoidal (right column) mode dominates the data set.

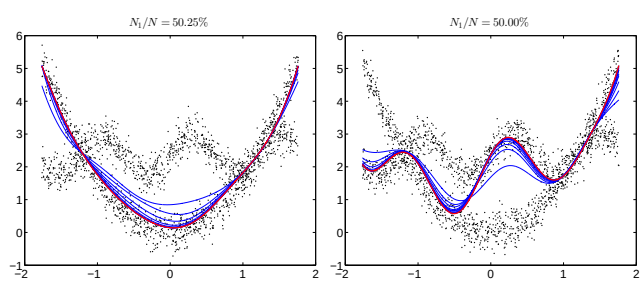

Figure 2: Iterations of the reweighting process for $N_1/N = 50.25\%$ (left) and $N_1/N = 50\%$ (right), where N_1 is the number of points generated by the quadratic.

arbitrarily switching between two modes as

$$y_i = \begin{cases} 0.9y_{i-1} + 0.2y_{i-2} + v_i, & \text{if } q_i = 1, \\ (0.8 - 0.5\exp(-y_{i-1}^2))y_{i-1} - \\ \quad (0.3 + 0.9\exp(-y_{i-1}^2))y_{i-2} + & \text{if } q_i = 2. \\ \quad 0.4\sin(2\pi y_{i-1}) + 0.4\sin(2\pi y_{i-2}) + v_i, \end{cases} \tag{20}$$

A training set of $N = 3000$ points is generated by (20) with a random sequence of q_i ($P(q_i = 1) = 2/3$ and $P(q_i = 2) = 1/3$), initial conditions $y_0 = y_{-1} = 0.1$, and an additive zero-mean Gaussian noise v_i of standard deviation $\sigma_v = 0.1$.

In this example, the linearity of the first mode is assumed to be known. Thus, a first submodel, f_1, is estimated with a linear kernel and $\varepsilon = 1.0043\sigma_v$, yielding the parameter estimates[5] reported in Table 2 (first column). Then, the points with $i \in I_R = \{i \ : \ |y_i - f_1(\boldsymbol{x}_i)| \leq 3\varepsilon\}$ are removed and a nonlinear submodel with a Gaussian RBF kernel ($\sigma = 0.5$) is estimated.

Similar experiments with the reversed order (nonlinear submodel estimated first) are also conducted on a data set with $P(q_i = 1) = 1/3$ (results in Table 2, second column).

The quality of the estimation is evaluated for each mode j in terms of the FIT criterion computed on a test set of 2000 data (generated without noise from the initial conditions $y_0 = 0.4,\ y_{-1} = -0.3$) as

$$\text{FIT}_j = \left(1 - \frac{\sqrt{\sum_{i \in I_j}(y_i - f_{q_i}(\boldsymbol{x}_i))^2}}{\sqrt{\sum_{i \in I_j}(y_i - m_j)^2}}\right),$$

where $I_j = \{i \ : \ q_i = j\}$ and m_j is the mean of y_i over all $i \in I_j$. Two additional performance indexes are used to evaluate the ability of the method to discriminate between the two modes during training: the fraction of data that must be removed and have been,

$$\text{D1} = \frac{|I_1 \cap I_R|}{|I_1|},$$

and the fraction of data that must be removed among those that have been,

$$\text{D2} = \frac{|I_1 \cap I_R|}{|I_R|}.$$

Note that these numbers are computed on the training data. The results, shown in Table 2, emphasize the accuracy of the estimated submodels and the fact that the proposed method correctly discriminates between the two modes, independently of the dominating mode.

Table 3 shows similar results for a switched NARX system with two nonlinear modes given by

$$y_i = \begin{cases} 0.4y_{i-1}^2 + 0.2y_{i-2}, & \text{if } q_i = 1, \\ (0.8 - 0.5\exp(-y_{i-1}^2))y_{i-1} - \\ \quad (0.3 + 0.9\exp(-y_{i-1}^2))y_{i-2} + & \text{if } q_i = 2. \\ \quad 0.4\sin(2\pi y_{i-1}) + 0.4\sin(2\pi y_{i-2}), \end{cases} \tag{21}$$

For this example, training trajectories of $N = 16000$ points are generated with 6000 points for mode 1 and 10000 points for mode 2 ($P(q_i = 1) = 0.375$). On these large-scale data sets, the average computing time was about one minute for each SVR training by LibSVM, i.e., for each iteration of the reweighted scheme.

[5]With a linear kernel, the parameters of a linear submodel (6) are recovered by $\boldsymbol{\theta} = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i$.

Table 2: Estimation of the system (20) switching between a linear mode (with parameters θ_1, θ_2) and a nonlinear mode. Numbers are averages and standard deviations over 100 trials with different noise, v_i, and mode, q_i, sequences.

$P(q_i = 1)$	2/3	1/3
θ_1 (= 0.9)	0.9008 ± 0.0092	0.9000 ± 0.0070
θ_2 (= 0.2)	0.1824 ± 0.0111	0.2019 ± 0.0068
FIT_1 (%)	97.9148 ± 0.8136	99.1472 ± 0.3261
FIT_2 (%)	83.7052 ± 5.3668	84.8237 ± 6.2603
D1 (%)	99.6840 ± 0.1383	98.7180 ± 0.4904
D2 (%)	88.4713 ± 0.6781	85.7150 ± 0.7850

Table 3: Estimation of the system (21). Numbers are averages and standard deviations over 100 trials with different noise, v_i, and mode, q_i, sequences.

FIT_1 (%)	73.2515 ± 4.2561
FIT_2 (%)	88.6945 ± 1.0960
D1 (%)	99.1160 ± 0.1618
D2 (%)	78.6506 ± 0.2756

6. CONCLUSIONS

This paper discussed the identification of hybrid systems involving arbitrary and unknown nonlinearities in the submodels, particularly focusing on the sparse optimization approach. Conditions of application of this approach were relaxed with the introduction of robust sparsity as a means to deal with noise in the data. We then emphasized the connections between this approach and the support vector machines developed in the field of machine learning. In particular, we have shown that nonlinear hybrid systems can be identified efficiently from large data sets by a sequence of SVM trainings. In addition, this formal equivalence allowed for the derivation of a modified algorithm for the automatic determination of the main hyperparameter (the threshold ε) in the robust sparsity approach. This modified algorithm introduces a new parameter, ν, which can be interpreted as the fraction of data considered as outliers for the model. The precise relationship between this parameter and the fraction of data generated by each mode, which is also involved in the sparse recovery conditions, is the subject of ongoing investigations. In particular, the characterization of the influence of the reweighting scheme on the choice of ν remains an open issue.

An alternative direction of future research concerns the computation of the full solution paths with respect to the regularization constant (λ) and the hyperparameters ε or ν. Here, the aim is to obtain the models for all possible values of the hyperparameters at a low computational cost. In this respect, we should once again take advantage of the equivalence with support vector regression and the large collection of results on this topic [11, 10, 37, 20].

The paper focused on systems with arbitrary switching mechanisms. While this provides an algorithm for a very general class of hybrid systems, this also implies that the active mode can only be predicted a posteriori (after the

observation of the actual output), which limits the applicability of the model for predictive purposes. In this respect, the proposed approach should be adapted to deal with piecewise smooth (PWS) systems, where the mode depends on a partition of the regression space. Meanwhile, the classical approach to this issue is to use a classification algorithm in a postprocessing step to recover the partition from the training points grouped into modes, as explained, e.g., in [24]. Note that SVMs were originally developed for classification and provide state-of-the-art solutions to such problems.

Finally, the concept of robust sparsity has also been introduced in the context of switched *linear* systems and more particularly for multi-input multi-output (MIMO) systems in state-space form [3].

7. REFERENCES

[1] L. Bako. Identification of switched linear systems via sparse optimization. *Automatica*, 47(4):668–677, 2011.

[2] L. Bako, K. Boukharouba, and S. Lecoeuche. An ℓ_0-ℓ_1 norm based optimization procedure for the identification of switched nonlinear systems. In *Proc. of the 49th IEEE Int. Conf. on Decision and Control (CDC)*, pages 4467–4472, 2010.

[3] L. Bako, V. L. Le, F. Lauer, and G. Bloch. Identification of MIMO switched state-space models. In *Proc. of the American Control Conference (ACC), Washington, DC, USA*, 2013.

[4] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10):1567–1580, 2005.

[5] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, Boston, 2004.

[6] L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors. *Large Scale Kernel Machines*. MIT Press, Cambridge, MA, 2007.

[7] E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008.

[8] C. Chang and C. Lin. LibSVM: a library for support vector machines, 2001. Available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm/.

[9] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.

[10] L. Gunter and J. Zhu. Computing the solution path for the regularized support vector regression. In *Advances in Neural Information Processing Systems 18*, pages 483–490, 2006.

[11] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

[12] T. Joachims. SVM^{light}, 1998. Available at http://svmlight.joachims.org/.

[13] A. L. Juloski, S. Weiland, and W. Heemels. A Bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50(10):1520–1533, 2005.

[14] J. Kwok and I. Tsang. Linear dependency between ε and the input noise in ε-support vector regression. *IEEE Transactions on Neural Networks*, 14(3):544–553, 2003.

[15] F. Lauer. Estimating the probability of success of a simple algorithm for switched linear regression. *Nonlinear Analysis: Hybrid Systems*, 8:31–47, 2013. Supplementary material available at `http://www.loria.fr/~lauer/klinreg/`.

[16] F. Lauer and G. Bloch. Switched and piecewise nonlinear hybrid system identification. In *Proc. of the 11th Int. Conf. on Hybrid Systems: Computation and Control (HSCC)*, volume 4981 of *LNCS*, pages 330–343, 2008.

[17] F. Lauer, G. Bloch, and R. Vidal. A continuous optimization framework for hybrid system identification. *Automatica*, 47(3):608–613, 2011.

[18] F. Lauer, V. L. Le, and G. Bloch. Learning smooth models of nonsmooth functions via convex optimization. In *Proc. of the IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP), Santander, Spain*, 2012.

[19] V. L. Le, G. Bloch, and F. Lauer. Reduced-size kernel models for nonlinear hybrid system identification. *IEEE Transactions on Neural Networks*, 22(12):2398–2405, 2011.

[20] G. Loosli, G. Gasso, and S. Canu. Regularization paths for ν-SVM and ν-SVR. In *Advances in Neural Networks – ISNN*, volume 4493 of *LNCS*, pages 486–496, 2007.

[21] O. Mangasarian and D. Musicant. Large scale kernel regression via linear programming. *Machine Learning*, 46(1):255–269, 2002.

[22] H. Ohlsson and L. Ljung. Identification of piecewise affine systems using sum-of-norms regularization. In *Proc. of the 18th IFAC World Congress*, pages 6640–6645, 2011.

[23] N. Ozay, M. Sznaier, C. Lagoa, and O. Camps. A sparsification approach to set membership identification of switched affine systems. *IEEE Transactions on Automatic Control*, 57(3):634–648, 2012.

[24] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal. Identification of hybrid systems: a tutorial. *European Journal of Control*, 13(2-3):242–262, 2007.

[25] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 185–208. MIT Press, 1999.

[26] J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37–50, 2004.

[27] B. Schölkopf, R. Herbrich, and A. Smola. A generalized representer theorem. In *Proc. of COLT/EuroCOLT*, volume 2111 of *LNAI*, pages 416–426, 2001.

[28] B. Schölkopf, A. Smola, R. Williamson, and P. Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.

[29] S. Shevade, S. Keerthi, C. Bhattacharyya, and K. Murthy. Improvements to the SMO algorithm for

SVM regression. *IEEE Transactions on Neural Networks*, 11(5):1188–1193, 2000.

[30] A. Smola, N. Murata, B. Schölkopf, and K. Müller. Asymptotically optimal choice of ε-loss for support vector machines. In *Proc. of the 8th Int. Conf. on Artificial Neural Networks (ICANN)*, pages 105–110, 1998.

[31] A. Smola, B. Schölkopf, and G. Rätsch. Linear programs for automatic accuracy control in regression. In *Proc. of the 9th Int. Conf. on Artificial Neural Networks (ICANN)*, pages 575–580, 1999.

[32] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[33] I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

[34] F. Tay and L. Cao. Modified support vector machines in financial time series forecasting. *Neurocomputing*, 48(1):847–861, 2002.

[35] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[36] R. Vidal, S. Soatto, Y. Ma, and S. Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proc. of the 42nd IEEE Conf. on Decision and Control (CDC)*, pages 167–172, 2003.

[37] G. Wang, D. Yeung, and F. Lochovsky. Two-dimensional solution path for support vector regression. In *Proc. of the 23rd Int. Conf. on Machine Learning (ICML)*, pages 993–1000, 2006.

APPENDIX

A. SKETCH OF PROOF OF THEOREM 1

Any function $f \in \mathcal{H}$ can be decomposed into a part in the span of $\{K(\boldsymbol{x}_i, \cdot) \ : \ \boldsymbol{x}_i \in \mathcal{D}_x\}$ and a part which is orthogonal to it, i.e., $f = u + v$, with, for all $\boldsymbol{x}_i \in \mathcal{D}_x$, $\langle K(\boldsymbol{x}_i, \cdot), v\rangle_{\mathcal{H}} = 0$. Thus, and by the reproducing property of K, for all $\boldsymbol{x}_i \in \mathcal{D}_x$, $f(\boldsymbol{x}_i) = \langle K(\boldsymbol{x}_i, \cdot), u + v\rangle_{\mathcal{H}} = \langle K(\boldsymbol{x}_i, \cdot), u\rangle_{\mathcal{H}} = u(\boldsymbol{x}_i)$. As a consequence, the first term in (2), which computes the error over the training set, does not depend on v. On the other hand, we have $\|f\|_{\mathcal{H}} = \|u + v\|_{\mathcal{H}} = \sqrt{\|u\|^2_{\mathcal{H}} + \|v\|^2_{\mathcal{H}} + 2\langle u, v\rangle_{\mathcal{H}}}$. Since $u \perp v$, $\langle u, v\rangle_{\mathcal{H}} = 0$. Thus, for any f with $v \neq 0$, there is a function $u \in Span\{K(\boldsymbol{x}_i, \cdot) \ : \ \boldsymbol{x}_i \in \mathcal{D}_x\}$ with smaller norm and which, in the conditions of Theorem 2.1, leads to a lower value of the objective function. A more detailed proof can be found in [27].

B. COMPLEXITY CONTROL VIA ℓ_1-NORM REGULARIZATION

Here, we show why the ℓ_1-norm regularization can be used in (7) instead of the RKHS norm to penalize the nonsmoothness of the model (i.e., the magnitude of the derivatives of f) and control its complexity. This can be seen from the fact that any f in the form used in (7) belongs to the RKHS $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$ and that its norm in this RKHS, given by (4), can be bounded by

$$\|f\|_{\mathcal{H}} = \sqrt{\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}} \leq \sqrt{\lambda_{\max} \|\boldsymbol{\alpha}\|_2^2} \leq \sqrt{\lambda_{\max}} \|\boldsymbol{\alpha}\|_1,$$

where $\lambda_{\max}$ is the largest eigenvalue of the Gram matrix $\boldsymbol{K}$ (as defined in Sect. 3.3.1). Thus, minimizing $\|\boldsymbol{\alpha}\|_1$ also provides control over $\|f\|_{\mathcal{H}}$ and the complexity of f.