

Nonlinear set-membership estimation: a support vector machine approach

Article

Published Version

Keesman, K. J. and Stappers, R. (2004) Nonlinear set-membership estimation: a support vector machine approach. Journal of inverse and ill-posed problems, 12 (1). pp. 27-42. ISSN 1569-3945 doi: 10.1515/156939404773972752 Available at <https://centaur.reading.ac.uk/28439/>

It is advisable to refer to the publisher's version if you intend to cite from the work. See [Guidance on citing](#).

To link to this article DOI: <http://dx.doi.org/10.1515/156939404773972752>

Publisher: De Gruyter

All outputs in CentAUR are protected by Intellectual Property Rights law, including copyright law. Copyright and IPR is retained by the creators or other copyright holders. Terms and conditions for use of this material are defined in the [End User Agreement](#).

www.reading.ac.uk/centaur

CentAUR

Central Archive at the University of Reading

Reading's research outputs online

Nonlinear Set-membership Estimation: A Support Vector Machine Approach

K.J. KEESMAN and R. STAPPERS *

Received August 30, 2003

Abstract

In this paper a support vector machine (SVM) approach for characterizing the feasible parameter set (FPS) in non-linear *set-membership* estimation problems is presented. It iteratively solves a regression problem from which an approximation of the boundary of the FPS can be determined. To guarantee convergence to the boundary the procedure includes a no-derivative line search and for an appropriate coverage of points on the FPS boundary it is suggested to start with a sequential box pavement procedure. The SVM approach is illustrated on a simple sine and exponential model with two parameters and an agro-forestry simulation model.

1 Introduction

Traditionally, given a set of input/output data the parameter estimation problem is solved by minimizing an objective function, typically the 2-norm of an error vector and sometimes augmented with a penalty term, under given constraints, *i.e.* the static or dynamic model structure. In this formulation the parameter vector ($\mathbf{x} \in \mathbb{R}^m$) which minimizes the objective function is considered to be the optimal parameter estimate.

Unlike this classical estimation approach, set-membership estimation is not concerned with minimizing an objective function. Instead of finding a single optimal parameter vector, a set of feasible parameters vectors, consistent with the model structure, measurements and *bounded* uncertainty characterization, will usually be found (see [1-3] for overviews).

*Systems and Control Group, Wageningen University, P.O. Box 43, 6700 AA Wageningen, The Netherlands. E-mail: karel.keesman@wur.nl.

Financial support from the European Union SAFE-project QLK5-CT-2001-00560.

The motivation for considering a set-membership approach is that in the traditional estimation problem formulation the characterization of the parameter uncertainty requires assumptions on mean, variances and/or probability density function of the errors. Especially in those cases where the model structure does not reflect the “real” system dynamics or when the available data set is limited the stochastic nature of the error sequence is questionable. The set-membership approach does not require a statistical description of the errors, instead it is assumed that the errors are *unknown-but-bounded*. For models linear in the parameters the resulting feasible parameter set (FPS) is a polytope. However, in *e.g.* ecological or economical applications the models are mostly nonlinear in their parameters. Then, the FPS is usually non-convex or even non-connected. Nevertheless, most non-linear set-membership approaches are able to give guaranteed inner or outer approximations for certain types of models.

Hence, the key issue in non-linear set-membership estimation is to find a suitable characterization of the FPS, which is easy to interpret and which tightly inner or outer bounds the FPS without being too computationally demanding. Typical approaches to the non-linear set-membership estimation problem are given by [4] using a discrete approximation method, by [5] using signomial programming and by [6-7] using interval analysis. See for further details the recent overview in [8].

The signomial programming approach, providing a tight outer-bounding box of the FPS, is most appropriate when dealing with exponential models. However, for a wide class of models of limited complexity the interval analysis approach is superior, because it gives guaranteed inner and outer bounds, in terms of a pavement of boxes, denoted as $[\mathbf{x}]_i$ where $i = 1, \dots, M$ with M the number of boxes, of the FPS at a prescribed accuracy. However, when the complexity of the model structure $\mathbf{F}(\mathbf{x})$ increases, the finding of a box enclosing function, denoted as $[\mathbf{F}](\mathbf{x})$, becomes problematic (see also remarks in [6]). First of all, results of interval computations depend on the way expressions are evaluated. *E.g.*, using the standard computational arithmetic rules, $[x][x]$ with $[x] \triangleq [-1, 1]$ leads to the interval $[-1, 1]$, while evaluating $[x^2]$ gives $[0, 1]$. Hence, especially for complex functions, it can be a quite laborious task to find an enclosing box. Moreover, the problem of finding the minimal enclosing box for complex models has not been solved, as yet. Secondly, and again for complex models, interval computation tends to a rather conservative way of error propagation due to the box-bounding error characterization and thus to a slow convergence of the algorithm.

Hence, when dealing with *complex simulation models* for which the internal structure is too complicated to be analyzed analytically via *e.g.* linearization or interval analysis, we have to rely on the input-output behavior of the model. A discrete approximation method using an appropriate sampling scheme may then be applied to gain insight into the set of feasible parameter vectors (see [4]). In this approach each sampled *feasible* parameter vector is consistent with the model, the measurements and the bounded error characterization and can thus

be seen as an unfalsified parameter vector. Note, furthermore, that in this case without knowledge of the internal structure of the model it is no longer possible to give guaranteed inner/outer bounds or to present the approximation error in the parameter space.

In this paper we aim at giving an *approximation* of the FPS boundary, where the approximation error is expressed in terms of a prediction error bound. This FPS boundary approximation can be subsequently used to improve the sample set from which new estimates in the neighborhood of the boundary are obtained. Hence, using prediction error bounds, the estimation accuracy of the algorithm is defined in the model output space instead of the parameter space and thus the identification objective is on model prediction quality and not so much on the identification of the exact FPS. However, usually it is not very appropriate when a large number of the feasible parameter vectors are situated in a small region in the neighborhood of the boundary. To avoid this situation a sequential box pavement approach will be used as an initial step to solve the set-membership estimation problem for complex model structures.

In section 2 we define the set-membership estimation problem. In section 3 the theory of support vector classification is explained. In section 4 the support vector regression problem is presented. In section 5 a method (i) to improve the estimate of the boundary of the FPS and (ii) to guarantee appropriate coverage in the parameter space is given. Three examples to illustrate the theory are presented in section 6. Finally, some concluding remarks are given.

2 Problem formulation

Consider the following nonlinear static (regression) model

$$\mathbf{y} = \mathbf{F}(\mathbf{x}) + \mathbf{e} \quad (2.1)$$

where $\mathbf{y} \in \mathbb{R}^N$ contains the observed output data, $\mathbf{F}(\mathbf{x})$ is a nonlinear vector function mapping the unknown parameter vector $\mathbf{x} \in \mathbb{R}^m$ into a noise-free model output $\hat{\mathbf{y}}$. Here we will assume that the weighted error vector $\mathbf{V}\mathbf{e} \in \mathbb{R}^N$ is bounded in the ℓ_∞ norm

$$\|\mathbf{e}\|_\infty^{\mathbf{V}} = \|\mathbf{V}\mathbf{e}\|_\infty \leq 1 \quad (2.2)$$

where \mathbf{V} is a diagonal weighting matrix.

The measurement uncertainty set (MUS), containing all possible output measurement vectors consistent with the observed output data and uncertainty characterizations is given by

$$\Omega_{\mathbf{y}} \triangleq \{\tilde{\mathbf{y}} \in \mathbb{R}^N : \|\mathbf{y} - \tilde{\mathbf{y}}\|_\infty^{\mathbf{V}} \leq 1\} \quad (2.3)$$

Let the feasible parameter set (FPS) be defined as

$$\Omega_{\mathbf{x}} \triangleq \{\mathbf{x} \in \mathbb{R}^m : \|\mathbf{y} - \mathbf{F}(\mathbf{x})\|_\infty^{\mathbf{V}} \leq 1\} \quad (2.4)$$

Then, the set-membership estimation problem is to characterize the feasible parameter set, which is consistent with the model (2.1), the data \mathbf{y} and the uncertainty characterization (2.2).

Alternatively, and in line with [6], we can define the error set

$$\Omega_{\mathbf{e}} \triangleq \{\mathbf{e} \in \mathbb{R}^N : \|\mathbf{e}\|_{\infty}^V \leq 1\} \quad (2.5)$$

Clearly, using (2.1) and (2.5), the FPS can also be defined as

$$\Omega_{\mathbf{x}} \triangleq \mathbf{F}^{-1}(\mathbf{y} - \Omega_{\mathbf{e}}) \quad (2.6)$$

where \mathbf{F}^{-1} is the reciprocal function, in set-theoretic sense, of \mathbf{F} . Hence, the problem can be seen as a *set inversion* problem.

The set-membership estimation approach presented in this paper starts from support vector classification (SVC), but finally uses support vector regression (SVR) to approximately characterize the boundary of the commonly non-convex or even non-connected FPS. In the next two sections a summary of these two statistical learning theories, within a set-membership context, is presented.

3 Classification using support vector machines

3.1 Linear separable case

Suppose we have dynamic simulation data $\{\mathbf{x}_i, \hat{\mathbf{y}}_i\}^\ell$, where the vector $\mathbf{x}_i \in \mathbb{R}^m$ is an element of the parameter space and the vector $\hat{\mathbf{y}}_i \in \mathbb{R}^N$ for $i = 1, \dots, \ell$ is the corresponding noise free model output.

For linear separable parameter vectors the support vector classifier selects the separating hyperplane

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b \quad (3.7)$$

such that

$$\begin{cases} \mathbf{w}^T \mathbf{x}_i + b \geq +1, & \text{if } \hat{\mathbf{y}}_i \in \Omega_y \\ \mathbf{w}^T \mathbf{x}_i + b \leq -1, & \text{if } \hat{\mathbf{y}}_i \notin \Omega_y \end{cases} \quad (3.8)$$

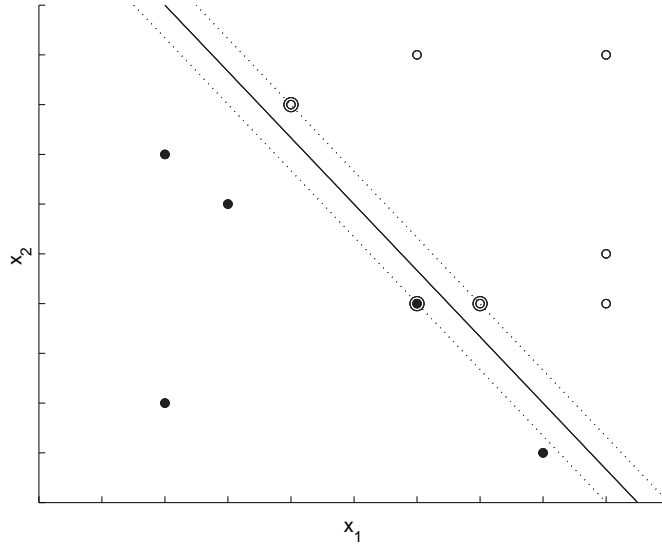
and such that the margin between the feasible and non-feasible parameter vectors is maximized (see Fig. 1 for an example with (\bullet) feasible and (\circ) non-feasible points).

Assign labels $z_i = +1$ if $\hat{\mathbf{y}}_i \in \Omega_y$ and $z_i = -1$ if $\hat{\mathbf{y}}_i \notin \Omega_y$. Then we can write the constraints (3.8) as

$$z_i[\mathbf{w}^T \mathbf{x}_i + b] - 1 \geq 0 \quad \forall i \quad (3.9)$$

The margin between the two hyperplanes $\mathbf{w}^T \mathbf{x} + b = 1$ and $\mathbf{w}^T \mathbf{x} + b = -1$ is equal to $2/\|\mathbf{w}\|$ [9]. Therefore maximizing the margin is equal to minimizing

$$\mathcal{J}(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}. \quad (3.10)$$


 Figure 1: Optimal linear separating hyperplane for $x \in \mathbb{R}^2$

Hence, to find the optimal hyperplane one has to solve the following quadratic optimization problem: minimize the quadratic form (3.10) subject to the linear constraints (3.9). In order to solve this quadratic optimization problem one has to find the saddle point of the Lagrange function.

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^{\ell} \alpha_i (z_i [\mathbf{w}^T \mathbf{x}_i + b] - 1) \quad (3.11)$$

where $\alpha_i \geq 0$ are the Lagrange multipliers. To find the saddle point one has to minimize this function over \mathbf{w} and b and to maximize it over the nonnegative Lagrange multipliers $\alpha_i \geq 0$ [10].

Setting the derivatives with respect to \mathbf{w} and b to zero gives

$$\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i z_i \mathbf{x}_i \quad (3.12)$$

$$\sum_{i=1}^{\ell} \alpha_i z_i = 0 \quad (3.13)$$

Substitution of (3.12) into (3.11) and taking into account (3.13) gives

$$W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j z_i z_j (\mathbf{x}_i \cdot \mathbf{x}_j) \quad (3.14)$$

where \cdot defines the dot product. Maximizing (3.14) is equivalent to minimizing (3.10).

To every parameter vector \mathbf{x}_i there belongs a Lagrange multiplier α_i . The parameter vectors \mathbf{x}_i for which $\alpha_i > 0$ and which uniquely determine the separating hyperplane (i.e. the points for which $z_i[\mathbf{w}^T \mathbf{x}_i + b] - 1 = 0$) and whose removal would change the solution are called *support vectors* (SV) and are indicating in Figure 1 with an extra circle. Parameter vectors for which $\alpha_i = 0$ indicate non-support vectors. Hence, parameter vectors close to the FPS boundary, and thus interesting for further analysis, can be easily detected on the basis of the associated values of α_i .

The separating hyperplane (3.7) can now be written as

$$f(\mathbf{x}) = \sum_{i \in SV} \alpha_i z_i \mathbf{x}_i \cdot \mathbf{x} + b \quad (3.15)$$

In set-membership estimation problems there will, in general, be no single linear separating hyperplane which satisfies the constraints (3.8). In the next section we will extend the ideas presented in this section to the nonlinear case.

3.2 Non-linear separable case

The quadratic problem in the previous section can be summarized as maximizing (3.14) subject to the constraint (3.13) and $\alpha_i \geq 0$, with the solution given by (3.15). Note that in this formulation the parameter vectors only appear as dot products. The idea is now to apply a nonlinear transformation $\varphi(\mathbf{x})$ on the parameter vectors and construct a linear separating hyperplane in the so-called *feature space* (possibly infinite dimensional). The classification is given by the sign of

$$f(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x}) + b \quad (3.16)$$

Using equation (3.15), we can rewrite (3.16) as

$$f(\mathbf{x}) = \sum_{i \in SV} \alpha_i z_i K(\mathbf{x}_i, \mathbf{x}) + b \quad (3.17)$$

where $K(\mathbf{x}_i, \mathbf{x}) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x})$. Hence, $f(\mathbf{x}) = 0$ describes the separating boundary in the parameter space. Typical choices for the kernel K are $K(\mathbf{x}_i, \mathbf{x}) = \mathbf{x}_i \cdot \mathbf{x}$ (linear SVM), $K(\mathbf{x}_i, \mathbf{x}) = (\mathbf{x}_i \cdot \mathbf{x} + 1)^d$ (polynomial SVM of degree d), $K(\mathbf{x}_i, \mathbf{x}) = \exp(-\|\mathbf{x}_i - \mathbf{x}\|_2^2 / \sigma^2)$ (Radial Basis Function SVM) and $K(\mathbf{x}_i, \mathbf{x}) = \tanh(\kappa \mathbf{x}_i \cdot \mathbf{x} + \theta)$. So instead of specifying the mapping $\mathbf{x} \rightarrow \varphi(\mathbf{x})$ we use the kernel to find the solution (see [10] for details).

The idea of mapping the parameters to a higher dimensional *feature space* can also be used for regression and will be briefly discussed in the next section.

4 Support vector regression

Suppose we have simulation data $\{\mathbf{x}_i, \varepsilon_i\}^\ell$ where the vector $\mathbf{x}_i \in \mathbb{R}^m$ is an element of the parameter space and the corresponding error bound is defined as

$$\varepsilon_i \triangleq \|\mathbf{e}_i\|_\infty^{\mathbf{V}} \quad (4.18)$$

where $\varepsilon_i \in \mathbb{R}$ is the weighted ∞ -norm of the errors for simulation i . In *feature space* the support vector regression, to allow a more informative and smoother description compared to the classification surface, is given by

$$\varepsilon(\mathbf{x}) = \mathbf{w}^T \varphi(\mathbf{x}) + b \quad (4.19)$$

In support vector regression again the objective function (3.10) is minimized but now subject to the equality constraints

$$\varepsilon_i = \mathbf{w}^T \varphi(\mathbf{x}_i) + b + \xi_i, \quad i = 1, \dots, \ell \quad (4.20)$$

Using the Lagrangian and the so-called *kernel trick* it can be shown [11] that the final regression is given by

$$\varepsilon(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \quad (4.21)$$

where α_i are Lagrange multipliers and K is a kernel function. Note that the α_i 's for the regression surface in (4.21) are not equal to the α_i 's for the classification surface in (3.17). Summarizing, starting with the linear classification problem we have arrived via nonlinear classification and support vector regression at Eqn. (4.21), which describes the weighted maximum error ε_i as a function of the parameter vector \mathbf{x} and which is starting point for further analysis in the next section.

5 Improving the bound

5.1 Sampling towards the boundary

Suppose we have found the optimal separating hyperplane for a training set $\{\mathbf{x}_i, \varepsilon_i\}^\ell$. The approximation of the FPS boundary using the regression surface $\varepsilon(\mathbf{x})$ is given by the set

$$\mathcal{B}_{\text{FPS}} = \{\mathbf{x} | \varepsilon(\mathbf{x}) = 1\} \quad (5.22)$$

(see Eqns. 2.2 and 4.18). In general, it is not possible to find an explicit solution \mathbf{x} for $\varepsilon(\mathbf{x}) = 1$.

We can approximate the regression surface in the neighborhood of \mathbf{x}_i using a first-order Taylor expansion

$$\varepsilon(\mathbf{x}_i + \delta) \approx l_i(\delta) = \varepsilon_i + \mathbf{g}_i^T \delta \quad \forall i \quad (5.23)$$

where $\varepsilon_i = \varepsilon(\mathbf{x}_i)$, \mathbf{g}_i is the gradient vector and $\delta = \mathbf{x} - \mathbf{x}_i$, so that l_i is the linear approximation in the neighborhood of \mathbf{x}_i . Instead of solving $\varepsilon(\mathbf{x}) = 1$, solving the linear rank deficient equation $l_i(\delta) = 1$ for δ gives the minimal 2-norm solution

$$\delta_{\text{LS}}^i = \mathbf{g}_i^{T\dagger} (1 - \varepsilon_i) \quad (5.24)$$

where $\mathbf{g}_i^{T\dagger} = \mathbf{g}_i(\mathbf{g}_i^T \mathbf{g}_i)^{-1}$ is the pseudo inverse of \mathbf{g}_i^T . Note that if $\mathbf{z} \in \text{null}(\mathbf{g}_i^T)$ then $\delta \triangleq \delta_{\text{LS}}^i + \mathbf{z}$ also solves $l_i(\delta) = 1$. The new parameter sample can now be obtained from $\mathbf{x}_i^{\text{new}} = \mathbf{x}_i + \delta_{\text{LS}}^i$. Taking the minimal 2-norm solution for δ corresponds to projecting the SV over the shortest distance on the approximation of the FPS boundary. Clearly, so far it is not guaranteed that this algorithm will converge to a point on the boundary of the FPS. Therefore, in addition to this gradient method using the *approximate* regression surface $\varepsilon(\mathbf{x})$, which may change when new points become available, a no-derivative line search method using solely the calculated error bounds ε_i related to points in the neighborhood of the FPS boundary is applied.

Let $\varepsilon(\mathbf{x})$ and $\varepsilon'(\mathbf{x})$ be continuous functions in \mathbf{x} and let us start with the kernel $K^{(j)}$, where j is an update index. The basic structure of the algorithm for finding the point \mathbf{x}_i^* on the FPS boundary, given the initial point $\mathbf{x}_i^{(k)}$ and the required convergence rate η_1 , is as follows:

Algorithm 1 *Boundary point detection*

step 1: set $k := k+1$ and obtain the new parameter sample $\mathbf{x}_i^{(k)} = \mathbf{x}_i^{(k-1)} + \delta_{\text{LS}}^i$ using the gradient method, Eqns. (5.23) and (5.24)

step 2: goto 1 and fix $K^{(j)}$ as long as $|1 - \varepsilon(\mathbf{x}_i^{(k-1)})| - |1 - \varepsilon(\mathbf{x}_i^{(k)})| > \eta_1$ else goto 3

step 3: perform a no-derivative line search using the two closest, in terms of the error bound ε_i , points to the boundary.

Notice that step 3 leads to the boundary point \mathbf{x}_i^* for which holds $\varepsilon(\mathbf{x}_i^*) = 1$, where the index i refers to a specific initial parameter vector in the parameter space. Conditions and rates of convergence for each of the steps 1 and 3 can be found in many textbooks on optimization (e.g. [12]). Hence, technical details are omitted here. In summary, under the mild condition that both $\varepsilon(\mathbf{x})$ and $\varepsilon'(\mathbf{x})$ are continuous functions in (\mathbf{x}) , which holds for a large class of models, a boundary point \mathbf{x}^* can be found. Note that in practice the line search will be aborted as soon as the error bound associated with the new estimate is close to 1, i.e. $|1 - \varepsilon(\mathbf{x}_i^{(k)})| < \eta_2$ with η_2 a small number indicating the boundary approximation error.

Hence, it is possible to find an ensemble of approximate boundary points of the FPS with weighted error bounds in the range $[1 - \eta_2, 1 + \eta_2]$. However, these are only local approximations of the FPS boundary.

5.2 Sampling for increased coverage

To obtain a better coverage of points on the boundary surface we propose to initialize the algorithm with a sequential box pavement approach. Define hereto the box $[\mathbf{x}]$ of \mathbb{R}^m , which is the cartesian product of m scalar intervals $[x] = [x^-, x^+] = \{x | x^- \leq x \leq x^+\}$. The width $w([\mathbf{x}])$ of the box $[\mathbf{x}]$ is given by $w([\mathbf{x}]) = \max_{j=1, \dots, m} \{x_j^+ - x_j^-\}$. Pre-specify, furthermore, the critical error bound ε^* or the critical width w^* and the required number of approximate boundary points I^* . Then, the algorithm proceeds as follows:

Algorithm 2 *Boundary point coverage*

step 1: set $k = 0$

step 2: define a regular rectangular grid in the parameter set $\Omega_x^{(k)}$

step 3: evaluate simulation model for each node $\mathbf{x}_i \in \Omega_x^{(k)}$ and determine ε_i (Eqn. 4.18)

step 4: set $k := k + 1$ and select those boxes $[\mathbf{x}_n]$ for $n = 1, 2, \dots$ which have at least one feasible vertex with associated error ε_i for $i = 1, 2, \dots$. These selected boxes form the new set $\Omega_x^{(k)}$

step 5: if $\max w([\mathbf{x}_n]) \leq w^$ for $n = 1, 2, \dots$ or if $|1 - \varepsilon_i| \leq \varepsilon^*$ for $i = 1, 2, \dots, I^*$ then stop else goto 2.*

If appropriate boxes have been found the vertices of these boxes can be further subject to the procedure of section 5.1, so that finally an ensemble of points in the pre-specified neighborhood of the FPS boundary with appropriate user-defined coverage can be found. Clearly, the SVM approach is more successful when the FPS boundary is smooth, as will be illustrated by the first example in the next section.

6 Examples

6.1 Sine function

In the first example we would like to show a one step procedure with a fine-meshed grid, thus without sequential box pavement and sequential sampling.

Consider, therefore, the regression model

$$y_i = \sin(x_1 t_i) + x_2 + e_i \quad (6.25)$$

where x_1 and x_2 are elements of the parameter vector. Suppose two measurements are given: for $t_1 = 1$; $y_1 \in [2.5, 3.5]$ and for $t_2 = 3$; $y_2 \in [0, 1]$. From the error bounds we derive the weighting matrix $\mathbf{V} = \text{diag}(\begin{bmatrix} 2 & 2 \end{bmatrix})$. Furthermore, assume that prior knowledge is available which bounds the feasible parameters by the intervals: $x_1 \in [0, 2\pi]$ and $x_2 \in [-0.5, 3]$.

In Figure 2 the exact and estimated (using Eqns. 5.22 and 4.21) boundary of the non-convex and non-connected FPS are given. The approximate boundary is directly determined from 225 parameter vector samples regular spaced on a 15×15 grid and subsequently processed by SVR providing α_i 's and b for a specific RBF kernel with $\sigma^2 = 0.12$. Notice the good approximation in this first step, except for $x_1 \in [2.7, 3.1]$, where the FPS boundary surfaces run almost parallel to each other. Clearly, in this region a finer grid is needed to properly describe the boundary.

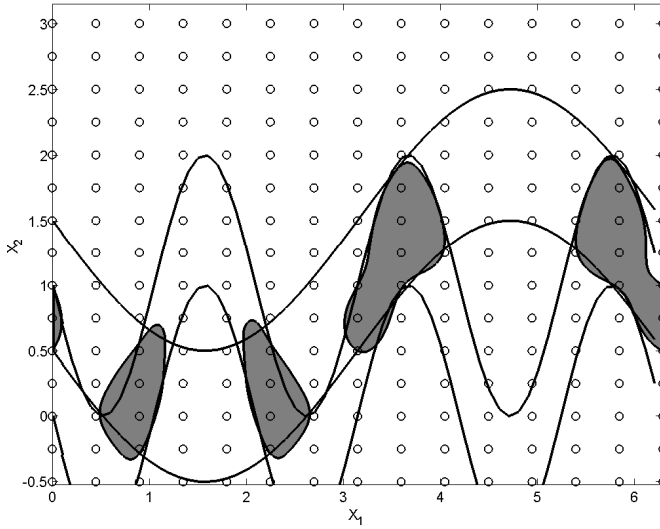


Figure 2: Estimated FPS boundary based on a 15×15 -grid.

6.2 Exponential function

Let us now illustrate the effects of the procedures described in section 5. Consider the non-linear regression model

$$y_i = x_2 \exp(-x_1 t_i) + e_i \quad (6.26)$$

where x_1 and x_2 are the model parameters. Suppose two measurements are given: for $t_1 = 1$: $y_1 \in [0.05, 0.25]$ and for $t_2 = 5$: $y_2 \in [0.02, 0.04]$. From the error bounds we derive the weighting matrix $\mathbf{V} = \text{diag}([10 \ 100])$. The minimum volume outer box using the signomial programming approach of [5] is given by the intervals: $0.0555 < x_1 < 0.6315$ and $0.0526 < x_2 < 0.4701$, which is taken here as an initial box for sampling.

In Figure 3 the exact and estimated boundary of the feasible parameter set are given. The approximate boundary is determined from 25 parameter vector

samples regularly spaced on a 5×5 grid (\otimes), which have been moved towards the boundary of the FPS using step 1 in Algorithm 1 ($*$) and where lines indicate the translation of the parameter vectors. Let us give some more details on this specific translation.

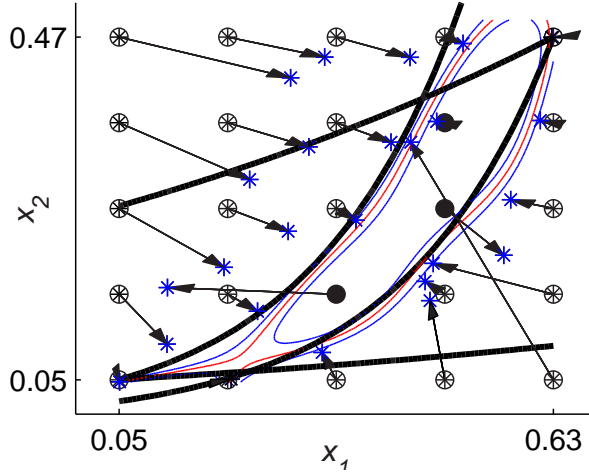


Figure 3: Estimated boundary based on RBF-SVM regression ($\sigma^2 = 0.022$).

Recall that the linear approximation of the regression surface in the neighborhood of \mathbf{x}_i is given by

$$l_i(\delta) = \varepsilon_i + \mathbf{g}_i^T \delta \quad (6.27)$$

For regressions based on the RBF-kernel the gradient vector \mathbf{g}_i is given by

$$\mathbf{g}_i = 2 \sum_{i=1}^{25} \frac{\alpha_i}{\sigma^2} (\mathbf{x} - \mathbf{x}_i) e^{-\|\mathbf{x} - \mathbf{x}_i\|_2^2 / \sigma^2} \quad (6.28)$$

The minimum norm update, as described in section 5, for the 25 sample parameter vectors is thus determined from (6.27) and (6.28).

The four thick lines define the exact boundary of the FPS. For the three approximate boundaries it holds that: $\varepsilon(x) = 1.1$ (outer-bound), 1.0 and 0.9 (inner-bound), respectively. Notice the effect that most points are moved in the direction of the boundary, except those which are already contained in the FPS. The reason for this is that the gradient of $\varepsilon(\mathbf{x})$ for $\mathbf{x} \in \Omega_x$ (FPS) is, in general, small, so that large errors in the calculation of δ_{LS}^i (Eqn. 5.24) may occur.

To obtain a better coverage near the vertices we propose to initialize the algorithm with a sequential box pavement approach, see Figure 4. Notice that starting with the coarse grid (5×5) in a larger prior parameter set $[0, 0.72] \times [0, 0.56]$ only two feasible parameter vectors are found (indicated by light-green

dots). In a second step, the grid of the boxes with these feasible points as vertices is refined, which gives another seven feasible points (dark-green dots). Consequently, in this first box pavement analysis nine feasible points have been found. The yellow shaded boxes indicate those boxes which may need further investigation for finding the boundary of the FPS, either using Algorithm 1 or using a further refinement of the shaded area.

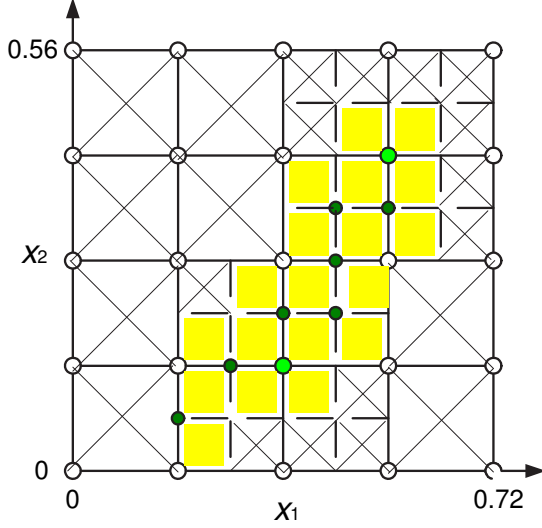


Figure 4: Box pavement approach.

6.3 Agro-forestry model

In the last example we focus on parameter estimation in a dynamic simulation model from limited data. Consider the following simplified agro-forestry model, which describes the growth of trees and crops, in terms of biomass and Leaf-Area-Index (LAI), under light competing conditions.

$$t \in [t_{bb}, t_{lf}] : \quad \frac{dLA_t}{dt} = \frac{1}{\tau_t} (LA_{sb}^{max} n_{Buds} - LA_t) \quad (6.29)$$

$$\frac{dB_t}{dt} = -K_{main} B_t + \frac{\varepsilon_t f_{it}}{\rho_t} I \quad (6.30)$$

$$t \in [t_{pl}, t_{hv}] : \quad \frac{dLAI_c}{dt} = p_l \varepsilon_c f_{ic} (1 - f_{it}) I s_c \quad (6.31)$$

$$\frac{dB_c}{dt} = \varepsilon_c f_{ic} (1 - f_{it}) I \quad (6.32)$$

$$p_t = \frac{\tau}{\tau + \overline{T}} p_{l_0} \quad (6.33)$$

with $f_{i_t} \triangleq 1 - e^{-k_t L A_t \rho_t}$ and $f_{i_c} \triangleq 1 - e^{-k_c L A I_c}$. Furthermore, t_{bb} is the time of bud-burst, t_{lf} of leaf fall, t_{pl} of crop planting and t_{hv} of crop harvesting. Notice that this is a very simple agro-forestry model (of nonlinear, hybrid, parameter-distributed system) with only four states, two inputs (radiation and temperature) and 10 parameters, *i.e.* τ_t , LA_{sb}^{max} , K_{main} , ε_t , ε_c , s_c , k_t , k_c , p_{l_0} and τ .

Let there be only one measurement of the crop yield, *i.e.* at harvest time $t = t_{hv}$: $B_c \in [1400, 1600] \text{ gm}^{-2}$, which is a very realistic situation in agro-forestry practice.

A preliminary sensitivity analysis revealed that among some other parameters LA_{sb}^{max} , the maximum potential leaf area of a single bud in m^{-2} , and ε_c , the potential crop growth rate in $\text{gMJ}^{-1}\text{m}^{-2}\text{day}^{-1}$, are parameters that are dominating the output behavior. Clearly, these two parameters can never be identified uniquely from the single measurement. Hence, an unbounded FPS is expected. The set-membership estimation result using SVM is shown in Figs. 5 and 6.

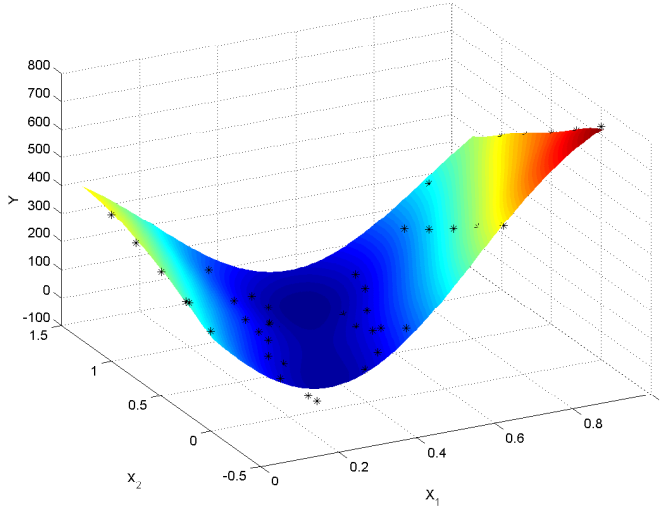


Figure 5: RBF-SVM regression with $x_1 = LA_{sb}^{max}$, $x_2 = \varepsilon_c$, $y = \varepsilon_i$ and $\sigma^2 = 0.5$

From the figures it appeared, however, that the SVM approach using a RBF kernel leads to a bounded set approximation of the FPS. Consequently, further evaluation of parameter vectors in the neighborhood of the boundary will reveal this incorrectness.

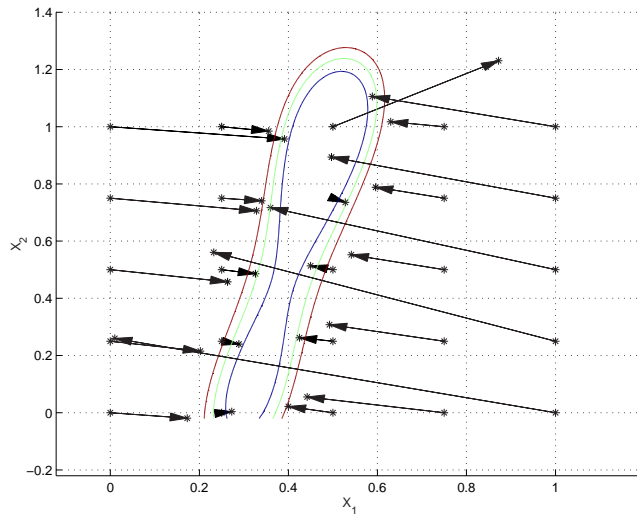


Figure 6: FPS boundary approximation with $\varepsilon(\mathbf{x}) = 1.1$ (red line), $\varepsilon(\mathbf{x}) = 1$ (green line) and $\varepsilon(\mathbf{x}) = 0.9$ (blue line)

7 Concluding remarks

In this paper nonlinear set-membership estimation using support vector machines (SVM) has been introduced. Set-membership estimation is most appropriate in case of small data sets. Using support vector regression (SVR) no initial feasible point is needed, so that the prior parameter set information can be limited. SVM together with bound improving algorithms is a valuable tool to effectively approximate FPS boundary point-wise.

The proposed method for solving the estimation problem for complex simulation models (of *e.g.* non-linear, hybrid systems) with bounded-noise data iteratively updates the estimate of the boundary by projecting a set of parameter vectors over the shortest distance towards the local approximation of the boundary followed by a no-derivative line search. Furthermore, additional application of a box pavement approach will avoid clustering of boundary points.

Because of the expected complex topology of the FPS for cases with more than 5-7 parameters, say, the method is most likely applicable to simulation models with a limited number of parameters. Hence, there is always a need to start with a sensitivity analysis to detect the dominant parameters!

REFERENCES

1. E. Walter (Ed.), Parameter identifications with error bounds. Special Issue *Math. Comp. Simul.*, (1990) **32(5&6)**.

2. M. Milanese, J.P. Norton, H. Piet-lahanier and E. Walter (Eds.). *Bounding Approaches to System Identification*, Plenum Press, NY (1996).
3. J. P. Norton (Ed.), Bounded-error estimation: issue 1&2. *Int. J. Adapt. Contr. and Sign. Process.*, (1994/1995) **8(1)** & **9(1)**.
4. K. J. Keesman, Membership-set estimation using random scanning and principal component analysis. *Math. Comp. Simul.* (1990) **32(5-6)**, 535–544.
5. M. Milanese, and A. Vicino, Estimation theory for nonlinear models and set membership uncertainty. *Automatica* (1991) **27(2)**, 403–408.
6. L. Jaulin, and E. Walter, Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica* (1993) **29(4)**, 1053–1064.
7. L. Jaulin, M. Kieffer, O. Didrit and E. Walter, *Applied Interval Analysis, With Examples in Parameter and State estimation, Robust Control and Robotics*. Springer, London (2001).
8. K. J. Keesman, Nonlinear-Model Case in Bound-based Identification, Article 6.43.11.2. In: *Control Systems, Robotics and Automation*, Ed. H. Unbehauen, in *Encyclopedia of Life Support Systems (EOLSS)*, Eolss Publishers, Oxford, UK (<http://www.eolss.net>) (2002).
9. C. J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* (1998) **2(2)**, 121–167.
10. V. Vapnik, *Statistical learning theory*. John Wiley & Sons, New York, 1998.
11. J. A.K. Suykens, L. Lukas, and J. Vandewalle, Sparse approximation using least squares support vector machines. In: *IEEE International Symposium on Circuits and Systems ISCAS'2000* (2000).
12. R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, New York, 1987.