Struct Multidisc Optim (2010) 42:517–529
DOI 10.1007/s00158-010-0511-0

RESEARCH PAPER

An improved adaptive sampling scheme for the construction of explicit boundaries

Anirban Basudhar · Samy Missoum

Received: 5 October 2009 / Revised: 23 March 2010 / Accepted: 26 March 2010 / Published online: 23 May 2010

Abstract This article presents an improved adaptive sampling scheme for the construction of explicit decision functions (constraints or limit state functions) using Support Vector Machines (SVMs). The proposed work presents substantial modifications to an earlier version of the scheme (Basudhar and Missoum, Comput Struct 86(19–20):1904–1917, 2008). The improvements consist of a different choice of samples, a more rigorous convergence criterion, and a new technique to select the SVM kernel parameters. Of particular interest is the choice of a new sample chosen to remove the "locking" of the SVM, a phenomenon that was not understood in the previous version of the algorithm. The new scheme is demonstrated on analytical problems of up to seven dimensions.

Keywords Support Vector Machines · Decision boundaries · Adaptive sampling

1 Introduction

In design optimization and uncertainty quantification, response surfaces and metamodels (Myers and Montgomery 2002; Wang and Shan 2007; Simpson et al. 2008) are some of the most commonly used approaches. Their purpose is to provide an approximation of an otherwise costly response from a computer simulation. Based on this approximation, often referred to as a surrogate, an optimization or calculation of a probability of failure can be efficiently performed (Helton 1993; Mourelatos et al. 2006; Queipo et al. 2008).

Among the limitations of the surrogate-based approaches, response discontinuities and the curse of dimensionality are the two most well known hurdles. In order to use expensive response evaluations in an intelligent manner and reduce computational costs, adaptive sampling techniques, such as Efficient Global Optimization (EGO) (Jones et al. 1998; Bichon et al. 2009) have gained popularity. A similar scheme for reliability analysis can be found in Bichon et al. (2007).

In order to tackle the discontinuity issue, the authors have developed an approach whereby the constraints of an optimization or the limit state function in a reliability problem are constructed explicitly in terms of the variables. That is, responses are no longer approximated such as in surrogate-based techniques. Therefore, the approach, referred to as explicit design space decomposition (EDSD) (Missoum et al. 2007; Basudhar et al. 2008), naturally handles discontinuities.

In the current version of EDSD, Support Vector Machines (SVMs) (Shawe-Taylor and Cristianini 2004; Tou and Gonzalez 1974; Vapnik 1998) are used to construct the explicit decision boundaries. The SVM-based EDSD has several inherent advantages that make it an attractive tool for optimization and probabilistic design (Basudhar et al. 2008; Basudhar and Missoum 2009a). Besides managing discontinuities, this approach enables the construction of highly nonlinear limit state functions and offers a simple way to propagate uncertainties (Hurtado 2004). Also, the technique can handle multiple failure modes simultaneously at no extra cost (Basudhar and Missoum 2009b).

In order to limit the number of samples required for an accurate approximation of SVM boundaries, an adaptive sampling technique was developed. Both serial as well as parallel update schemes were implemented for the selection of samples. The scheme was successfully applied to problems up to five variables (Basudhar and Missoum 2007,

A. Basudhar · S. Missoum (✉)
Aerospace and Mechanical Engineering Department,
The University of Arizona, Tucson, AZ 85721, USA
e-mail: smissoum@ame.arizona.edu

2008). However, certain aspects of the update algorithm were later understood that limited its application to problems with higher dimensionality.

In this article, an improved adaptive sampling technique is proposed for refining SVM boundaries. The proposed update scheme differs from the previous approach in three major areas:

- *Selection of samples:* The samples chosen during the adaptive scheme to refine the explicit boundary are now defined through two optimization problems. Of particular interest is the choice of a sample whose purpose is to remove potential locking of the SVM, a phenomenon that was not understood in the first version of the adaptive scheme (Basudhar and Missoum 2008).
- *Convergence measure:* The new convergence measure is based on the comparison of coefficients of the polynomial SVM between two successive iterations. This approach is more rigorous and scalable than the previous method which involved a very large (non-scalable) number of so-called convergence points.
- *Selection of SVM parameters:* The algorithm uses a polynomial kernel function to define the SVM boundaries instead of a Gaussian radial basis function. The polynomial degree is automatically selected and modified. In the previous scheme, fixed parameter values of the kernel were used.

In order to demonstrate the efficacy of the proposed algorithm, analytical examples are presented with up to seven variables. The evolution of the convergence and error measures as a function of the number of samples is presented. The actual functions to be approximated are derived from a single general formula, function of dimensionality. This was done with the purpose of having comparable levels of difficulty for all the examples.

The remainder of this paper is constructed as follows: in Section 2, a short introduction to SVM is given. The previous EDSD adaptive sampling methodology is explained in Section 3. Section 4 gives a summary of the limitations of the previous scheme. This is followed by a section explaining the new methodology. Finally, the examples are presented in Section 6.

2 Support Vector Machines (SVMs)

An SVM is a machine learning technique used for the classification of data. It has the ability to explicitly define multidimensional and complex boundaries that optimally separate two classes of data (Shawe-Taylor and Cristianini 2004; Tou and Gonzalez 1974; Vapnik 1998; Gunn 1998). These features make it a natural choice for the construction of constraints or limit state functions (Section 3).

The purpose of this section is to give a brief overview of the SVM classification methodology and expose the reader to the SVM jargon. Consider a d-dimensional space sampled with N training points $\mathbf{x}_i$. Each point is associated with one of two classes characterized by a value $y_i = \pm 1$. The SVM algorithm finds the boundary that optimally separates the two classes of data samples. The corresponding boundary equation is:

$$s(\mathbf{x}) = b + \sum_{i=1}^{N} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) = 0 \tag{1}$$

where b is a scalar referred to as the bias, λ_i are Lagrange multipliers obtained from the quadratic programming optimization problem used to construct the SVM. K is the kernel of the SVM. With (1), the classification of any arbitrary point $\mathbf{x}$ is given by the sign of s.

The optimization problem used to solve for the optimal SVM classifier involves the maximization of a "margin". The notion of margin can be simply understood by inspecting a linear SVM classifier (Fig. 1). In this case, the margin is the distance between two parallel hyperplanes, referred to as support hyperplanes, given by $s(\mathbf{x}) = \pm 1$ in the space of input variables $\mathbf{x}_i$. These support hyperplanes go through

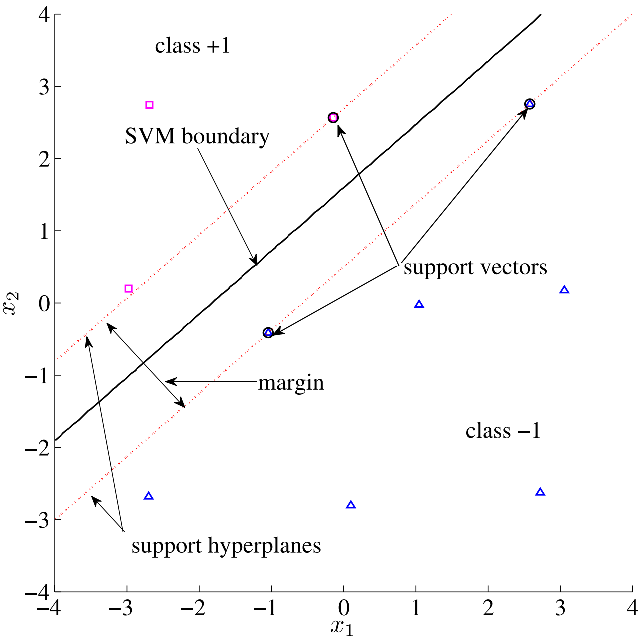

Fig. 1 Linear SVM classifier separating class +1 from class −1. The margin is the distance between the *dashed lines* which are the support hyperplanes. The support vectors lying on the support hyperplanes are shown by *black circles*

one or several training samples referred to as support vectors. In addition to the maximization of the margin, the SVM optimization problem is subjected to a constraint enforcing that the margin space should not contain any training sample.

The Lagrange multipliers associated with the support vectors are positive while the other samples have Lagrange multipliers equal to zero. That is, an SVM classifier trained using only the support vectors is identical as the one obtained using all the data samples. Typically, the number of support vectors is much smaller than *N*.

The SVM boundary can be linear or highly nonlinear. The kernel *K* in the SVM equation can have different forms such as polynomial, Gaussian radial basis function, multilayer perceptron etc. An interesting feature that facilitates the construction of an SVM is that there is always a higher dimensional space (the feature space) where the boundary is linear.

The previous EDSD studies with SVM (Basudhar and Missoum 2008; Basudhar et al. 2008) used Gaussian radial basis functions. In this article, a polynomial kernel is used. The motivation for this choice is explained in Section 5.4. A polynomial kernel is given as:

$$K(\mathbf{x}_i, \mathbf{x}) = (1 + \langle \mathbf{x}_i, \mathbf{x} \rangle)^p \qquad (2)$$

where *p* is the degree of the polynomial kernel. An example of SVM classification with a polynomial kernel is depicted in Fig. 2.

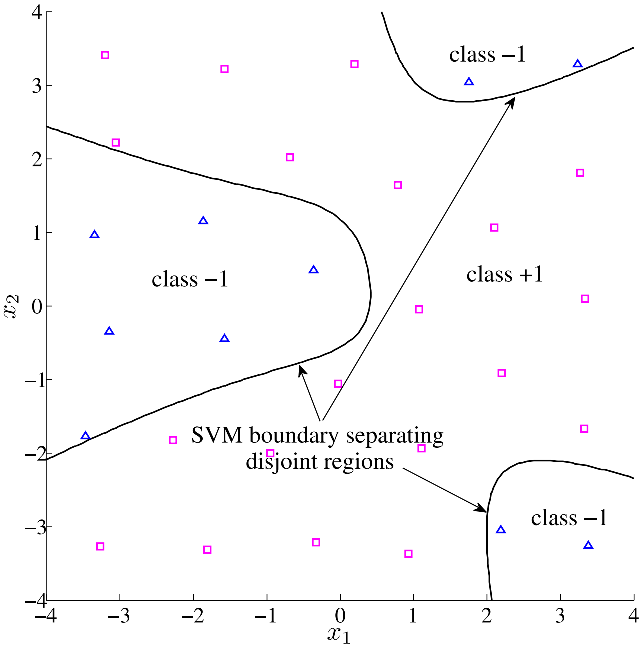

Fig. 2 SVM boundary based on a polynomial kernel separating disjoint classes (+1 and −1)

3 Previous methodology for the identification of explicit boundaries

The previous work on the construction of explicit boundaries is presented in this section. A brief overview of the basic EDSD approach and a previous adaptive sampling scheme is presented (Basudhar and Missoum 2008; Basudhar et al. 2008). The limitations of the previous scheme are identified and a new adaptive sampling scheme is described.

3.1 Basic EDSD methodology with a fixed design of experiments

The main idea behind EDSD (Basudhar et al. 2008) is to solve a classification problem and divide the space into regions corresponding to specific states of a system. The boundary separating the distinct regions can be used as a limit state function or an optimization constraint. In order to construct the classifier, the space is first sampled using a uniform design of experiments (DOE) such as Improved Distributed Hypercube Sampling (IHS) (Beachkofski and Grandhi 2002) or Centroidal Voronoi Tessellations (CVT) (Romero et al. 2006). These samples are then classified based on the corresponding response values, followed by the construction of a boundary that separates the distinct classes of samples. The use of SVMs for the definition of explicit boundaries has been found to be flexible due to their ability to represent highly nonlinear boundaries and disjoint regions (Fig. 2) (Basudhar et al. 2008). The main steps of EDSD using an SVM classifier are listed in Algorithm 1.

3.2 Previous adaptive sampling scheme for the construction of boundaries

The use of SVM for EDSD was first proposed in order to handle the probabilistic design of problems with nonlinear limit state functions and response discontinuities (Basudhar et al. 2008). Subsequently, an update scheme was developed which reduced the number of samples required to

Algorithm 1 Basic EDSD methodology using an SVM classifier

```
1: Sample the space with a CVT DOE.
2: Evaluate the system response at each sample (e.g. using a finite element code).
3: Classify the samples into two classes (e.g. safe and failed) based on the response values. The classification is performed using a threshold value or a clustering technique if the response are discontinuous (Basudhar et al. 2008).
4: Construct the SVM classifier.
```

construct an accurate decision function (Basudhar and Missoum 2008). A relatively small DOE was used to construct the initial SVM boundary. It was then refined with samples selected based on the following criteria:

- The samples were selected in regions with the highest probability of misclassification. Such regions are identified as the ones lying on the SVM boundary.
- The samples were selected in sparsely populated regions of the space.

A convergence measure was obtained by quantifying the change in SVM between two iterations. For this purpose, a set of convergence points was defined over the entire space. The change was quantified as the fraction of convergence points with a change of predicted class by the SVM between two consecutive iterations.

4 Limitations of the previous adaptive sampling scheme

The previous adaptive sampling technique was a major improvement over the first EDSD approach with a fixed DOE. That is, the number of samples to achieve a certain accuracy of the decision function was reduced. However, certain limitations of the methodology, object of this article, were gradually realized:

- *The sample selection sequence:* The previous algorithm consisted of three sample selection steps. The first step involved the selection of a sample on the boundary with a constraint on the minimum allowable distance to an existing sample. In the second step, a sample was selected by maximizing the distance to the previous sample. This sample was constrained to lie on the SVM boundary and to be located at a minimum distance from any existing sample. However, the first step did not provide a unique solution. Further, steps 1 and 2 (Basudhar and Missoum 2008) involved arbitrary coefficients for defining the minimum allowable distance to existing samples. For these reasons, these two steps do not seem to be justified unlike the third step, which is the basis of the new proposed update scheme (Section 5.1). This step locates a sample on the boundary so as to maximize the minimum distance to existing samples.
- *"Locking" of the SVM:* The previous scheme involved the selection of new samples only on the SVM boundary. The motivation was that these samples have the largest "probability of misclassification". Also, it was reasoned that since a sample selected on the boundary lies in the margin of SVM, such a sample compels the boundary to be modified, thus adding useful information to the problem. However, it was later realized that the modification of SVM boundary due to such a sample may be negligible if the margin (loosely, the local distance between $s(\mathbf{x}) = +1$ and $s(\mathbf{x}) = -1$) is thin, thus wasting function evaluations. When locating a sample on the SVM within a thin margin, which by construction should not contain any sample, the change

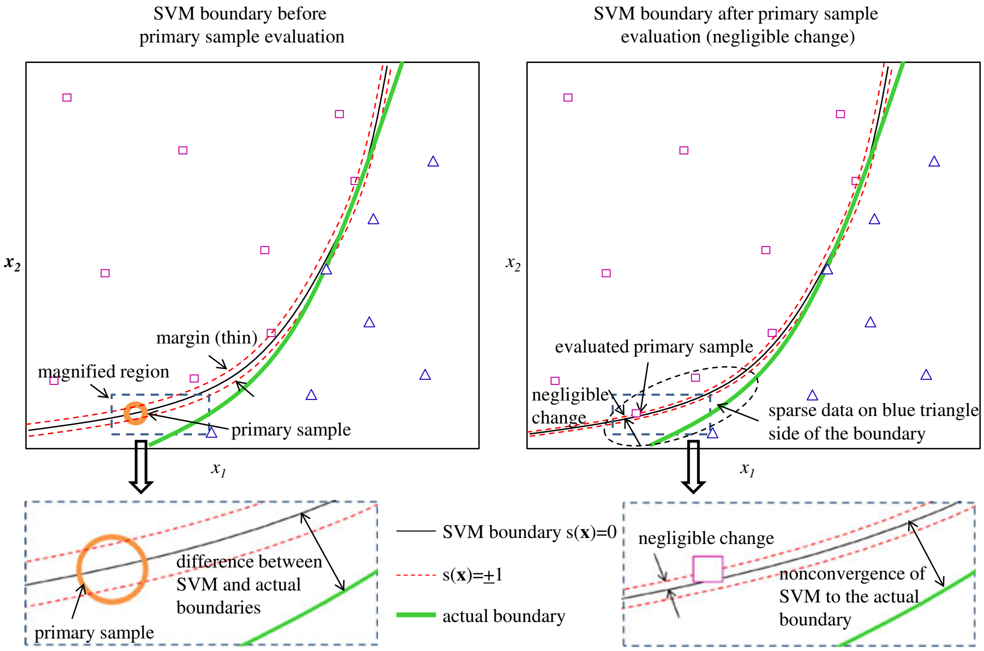
SVM boundary before
primary sample evaluation
SVM boundary after primary sample
evaluation (negligible change)
x_2
x_1
margin (thin)
magnified region
primary sample
evaluated primary sample
negligible change
sparse data on blue triangle side of the boundary
difference between SVM and actual boundaries
primary sample
SVM boundary s(**x**)=0
s(**x**)=±1
actual boundary
negligible change
nonconvergence of SVM to the actual boundary

Fig. 3 Locking of the SVM due to reduction of the margin (region between *dashed red curves*). A new sample on the boundary (*brown circle*), belonging to the *magenta square* class, produces negligible change although there is no sample belonging to the opposite class (*blue triangles*) nearby

in boundary due to the update is inevitably small. If this small change occurs in a region with a relatively uniform amount of information from both classes in the vicinity of the added sample, then the SVM can be assumed to be locally accurate. However, if the data from one class is sparse, then the low convergence rate becomes an issue. This is referred to as the "locking" of the SVM (Fig. 3). As a result of the locking phenomenon, the SVM boundary may not converge to the actual one with reasonable amount of data in certain localized regions. In addition to primary samples selected on the SVM boundary, a secondary sample (see Section 5.2) directed specifically at the prevention of SVM locking is evaluated in the new proposed algorithm (Fig. 5).

- *Scalability of the convergence measure:* The convergence measure in the previous scheme required a large number of convergence points in the space. The change was quantified as the fraction of these convergence points that switched class between two consecutive iterations based on the respective SVM boundaries. However, accuracy of the convergence measure depended on the number of convergence points that were defined. Since a large number of points was required, the approach was not scalable to high dimensional problems. A new, more rigorous, convergence measure is based on the comparison of coefficients of the polynomial SVM equations between successive iterations. This approach is scalable to high dimensional problems.
- *SVM parameters were fixed:* The kernel parameters for constructing the SVM boundary were fixed at the start of the previous scheme. In the literature, selection of kernel parameters is usually performed using cross-validation (Cawley and Talbot 2003). In this paper, a method to automatically select the polynomial kernel parameters during the update is presented (Section 5.3).

5 Improved adaptive sampling methodology

The new scheme developed to overcome the aforementioned limitations associated with the previous adaptive sampling scheme is presented in this section. The summary of the methodology for constructing the boundaries is presented in Algorithm 2. For the sake of clarity, the details of the scheme are presented in the subsequent sections.

Algorithm 2 Methodology for the construction of explicit boundaries

```
1:  Sample the space with a CVT DOE.
2:  Evaluate the system response at each sample (e.g. using a finite element code).
3:  Classify the samples into two classes (e.g. safe and failed) based on the response values. The classification is performed using a threshold value or a clustering technique.
4:  Set iteration $k = 0$
5:  Select the parameters for constructing the SVM boundary (Section 5.3).
6:  Construct the initial SVM boundary that separates the classified samples.
7:  repeat
8:      $k = k + 1$
9:      Select a primary sample on the SVM boundary (Section 5.1) and reconstruct the SVM with the new information.
10:     Repeat 9 to select another primary sample.
11:     Select a secondary sample to prevent locking of the SVM (Section 5.2). Reconstruct the SVM boundary.
12:     Modify the SVM parameters if any of the training samples are misclassified (Section 5.3). Reconstruct the SVM boundary.
13:     Calculate the convergence measure $\Delta_k$.
14: until $\Delta_k \leq \delta_1$
```

5.1 Selection of primary samples on the SVM boundary

At each iteration, two primary samples and one secondary sample are evaluated. The selection of a primary sample is performed by maximizing the distance to the closest training sample while lying on the SVM boundary (4). As mentioned in Section 3.2, a sample on the boundary has the highest probability of misclassification. Also, such a sample lies in the margin of SVM and, therefore, compels the boundary to change. The selection of samples in sparsely populated regions avoids redundancy of data. Figure 4 shows the selection of a primary sample and the SVM boundary update due to it. The optimization problem to select a primary sample is:

$$\begin{aligned} \max_{\mathbf{x}} \quad & ||\mathbf{x} - \mathbf{x}_{nearest}|| \\ s.t. \quad & s(\mathbf{x}) = 0 \end{aligned} \tag{3}$$

where $\mathbf{x}_{nearest}$ is the training sample closest to the new sample. This is a "maxmin" problem for which the objective function is non-differentiable. The problem is made differentiable by reformulating it as:

$$\begin{aligned} \max_{\mathbf{x},z} \quad & z \\ s.t. \quad & ||\mathbf{x} - \mathbf{x}_i|| \geq z \quad i = 1, 2, ..., N \\ & s(\mathbf{x}) = 0 \end{aligned} \tag{4}$$

In this work, the differentiable formulation of the global optimization problem is solved using a local optimizer

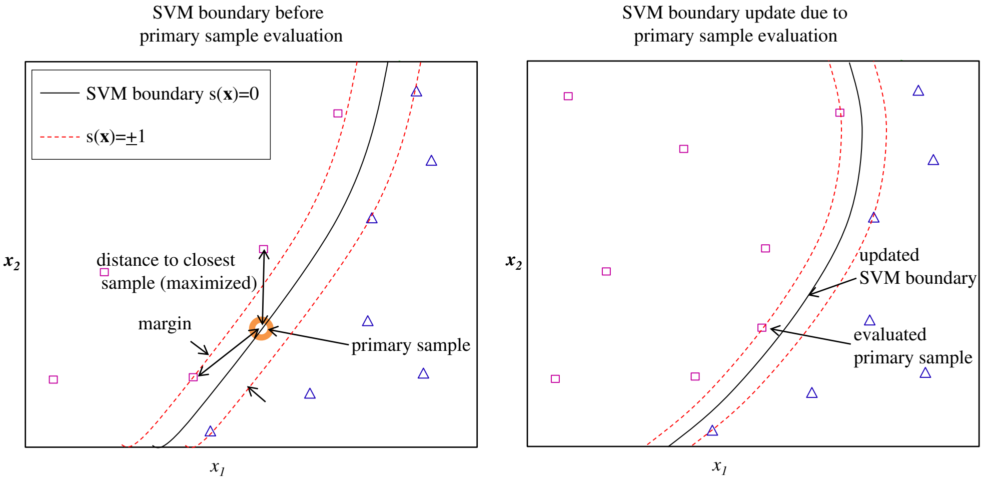

Fig. 4 Selection of a new training sample on the SVM boundary while maximizing the distance to the closest sample (*left*). Updated SVM decision boundary (*right*)

(sequential quadratic programming) with multiple starting points given by the existing training samples.

5.2 Selection of a secondary sample to prevent locking of the SVM

As pointed out in Section 4, the rate of convergence of the SVM boundary to the actual one may be very slow due to the SVM locking phenomenon (Fig. 3). The selection of a secondary sample to prevent the locking of the SVM is described in this section.

The secondary sample is aimed at removing the locking by positioning a sample in a region where data from one class is sparse in the vicinity of the boundary. If this sample is misclassified by the current SVM, this might lead to significant change of the SVM boundary (Fig. 5). The selection of the sample is a two step process:

- The support vector $\mathbf{x}^*_{sv}$ farthest from existing samples of opposite class is identified. A hypersphere of radius R centered around the support vector is then defined. R is chosen as half the distance from $\mathbf{x}^*_{sv}$ to the closest sample $\mathbf{x}_{opp}$ belonging to the opposite class:

$$R = \frac{1}{2}\left|\left|\mathbf{x}^*_{sv} - \mathbf{x}_{opp}\right|\right| \tag{5}$$

- The secondary sample is selected within the hypersphere so that it belongs to the opposite class of $\mathbf{x}^*_{sv}$ according to the current SVM prediction:

$$\begin{aligned} \min_{\mathbf{x}} \quad & s(\mathbf{x})y^*_{sv} \\ s.t. \quad & \left|\left|\mathbf{x} - \mathbf{x}^*_{sv}\right|\right| \le R \\ & s(\mathbf{x})y^*_{sv} \le 0 \end{aligned} \tag{6}$$

where y^*_{sv} is the class label (±1) of the selected support vector $\mathbf{x}^*_{sv}$. The objective function in (6) also appears as a constraint in order to avoid an optimum solution with a positive objective function value, i.e. to avoid a solution for which the current SVM provides the same class

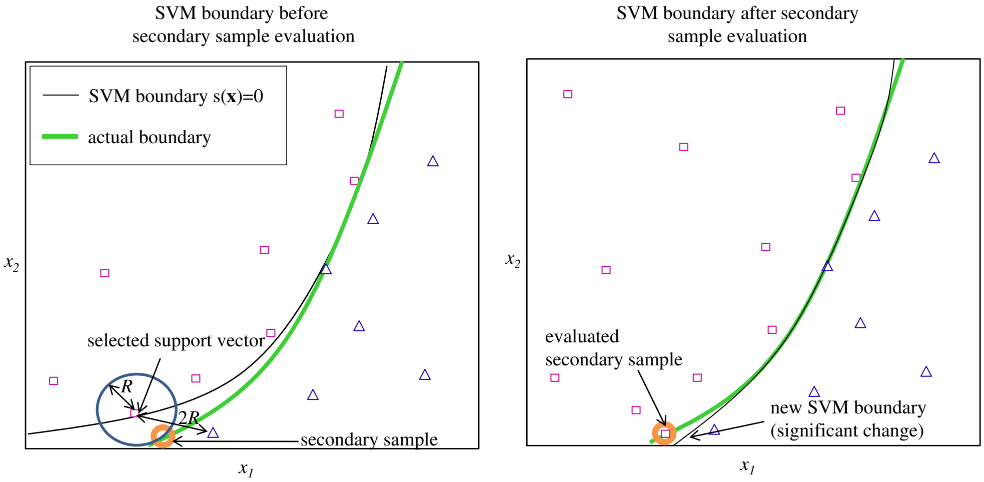

Fig. 5 Evaluation of a secondary sample to prevent locking of the SVM boundary

for the support vector and the secondary sample. In the cases where the problem is infeasible, a primary sample is added (4). Also, it is noteworthy that the choice of R as one half of $\left|\left|\mathbf{x}_{sv}^{*} - \mathbf{x}_{opp}\right|\right|$ will prevent the sample to be chosen too close to regions with existing samples such as $\mathbf{x}_{opp}$ itself.

5.3 Selection of kernel parameters

The selection of kernel parameters is of prime importance for the construction of the SVM decision boundary. Several studies in the computer science community use cross-validation techniques for the selection of kernel parameters (Cawley and Talbot 2003). In this article, cross-validation is not used. However, the SVM parameters are selected such that the boundary constructed is the "simplest" one without any training sample misclassification. For the polynomial kernel used in this study, this corresponds to the lowest degree polynomial, which does not produce any training misclassification (misclassification of already evaluated training samples).

5.4 Convergence criterion

Because the actual function is not known in general, the convergence criterion for the update algorithm is based on the variation of the approximated SVM boundary between two consecutive iterations. Since the polynomial kernel is used, a rigorous quantification of the variation is possible based on the coefficients. Unlike the previous scheme (Basudhar and Missoum 2008), this does not require a large number of "convergence" points. In order to compare the polynomials at iterations $k-1$ and k, the coefficients are scaled such that the largest coefficient (absolute value) at iteration $k-1$ is 1. The corresponding coefficient for iteration k is also set to 1. The calculation of the convergence measure is implemented as follows:

- *Find the polynomial coefficients:* In order to find the polynomial coefficients, a linear system of equations is solved. For a d-dimensional problem and a polynomial kernel of degree p, the number of coefficients is $\binom{d+p}{p}$. In order to find the coefficients, a set of $\binom{d+p}{p}$ points is selected from a CVT distribution and the corresponding SVM values are calculated. The coefficients are obtained as:

  $$\boldsymbol{\alpha} = \mathbf{Q}^{-1}\mathbf{s} \tag{7}$$

  where $\mathbf{s}$ is the array of SVM values. The i^{th} row of the matrix $\mathbf{Q}$ is given as:

  $$\mathbf{R}_i = \left(1 \; x_1 \; x_2 \; \ldots \; x_d \; \ldots \; \ldots \; x_1^p \; \left(x_1^{p-1} x_2\right) \ldots \; x_d^p\right)\Big|_{\mathbf{x}_i} \tag{8}$$

  Thus, the matrix $\mathbf{Q}$ is a square matrix of size $\binom{d+p}{p} \times \binom{d+p}{p}$. Note that the matrix $\mathbf{Q}$ is invertible and well conditioned as the samples to construct it are uniformly distributed with CVT.

  The coefficients could also be calculated using multinomial expansion (Ma 2001) and (1). The coefficient of a general term $x_1^{p_1} x_2^{p_2} \ldots x_d^{p_d}$, except for the constant term, in the SVM equation is given as:

  $$\alpha_{p_1 p_2 \ldots p_d} = \frac{p!}{\prod_{j=0}^{d} p_j!} \sum_{i=1}^{N} \left(\lambda_i y_i \prod_{j=1}^{d} x_j^{p_j}\right)\Bigg|_{\mathbf{x}_i}$$

  $$\text{where} \quad \sum_{j=0}^{d} p_j = p \tag{9}$$

  The constant term in the SVM equation is equal to b (1).
- *Comparison of the coefficients between iterations:* In order to compare the coefficients between successive iterations $k-1$ and k, the coefficients corresponding to different degrees are separated into distinct arrays. The array of coefficients corresponding to degree m is denoted as $\boldsymbol{\alpha}_m$. The evolution of the coefficients is studied separately for each degree (Fig. 8). The reason for studying each degree separately is that an identical relative change for two coefficients, especially for the largest and smallest degrees, may not lead to the same change in the boundary. The relative change in the norm of $\boldsymbol{\alpha}_m$ is calculated for each degree and the maximum value is used as a measure of convergence. The convergence measure is given by:

  $$\Delta_k = \max_m \left(\Delta_k^{(m)}\right) \tag{10}$$

  where $\Delta_k^{(m)}$ is given as:

  $$\Delta_k^{(m)} = \frac{\left|\left|\boldsymbol{\alpha}_m^{(k)} - \boldsymbol{\alpha}_m^{(k-1)}\right|\right|}{\left|\left|\boldsymbol{\alpha}_m^{(k-1)}\right|\right|} \tag{11}$$

5.5 Error measures

The accuracy of an approximated SVM boundary is judged by its fidelity to the actual function. In practical problems, an error metric may not be available. However, error measures can be obtained in the case of academic analytical test functions. Two distinct error metrics are presented:

- *Based on "test" points:* The error may be quantified as the fraction of the spatial volume which is misclassified by the SVM boundary. For this purpose, a set of N_{test} randomly generated "test" points is used to densely sample the whole space. The values of both the actual function and the SVM are calculated for each test point. Since the actual function is analytical, these function evaluations are efficiently performed. The number of test points being much larger than the number of sample points, the error can be assessed by calculating the fraction of misclassified test points (Basudhar and Missoum 2008). A test point for which the SVM and the actual function provide different signs is considered misclassified. The error ϵ_k is given below:

$$\epsilon_k = \frac{num\,(s(\mathbf{x}_{test})y_{test} \leq 0)}{N_{test}} \tag{12}$$

where $\mathbf{x}_{test}$ and y_{test} represent a test sample and the corresponding class value (± 1) for the actual (known) decision function.

- *Based on polynomial coefficients of the SVM boundary:* ϵ_k is a good measure of the fraction of misclassified space if the space is sampled densely. However, the approach is limited to a few dimensions due to constraints on computational resources. Fortunately, a measure based on polynomial coefficients is possible for actual decision boundaries represented by polynomials. The relative error E_k is given by:

$$E_k = \frac{\left|\left|\boldsymbol{\alpha}^{(act)} - \boldsymbol{\alpha}^{(k)}\right|\right|}{\left|\left|\boldsymbol{\alpha}^{(act)}\right|\right|} \tag{13}$$

where $\boldsymbol{\alpha}^{(act)}$ is the array of the polynomial coefficients for the actual function.

6 Examples

Several test examples demonstrating the ability of the update methodology to reconstruct known analytical functions are presented. The analytical decision functions are written in the form $f(\mathbf{x}) = 0$. In order to perform the SVM classification, the samples corresponding to $f(\mathbf{x}) > 0$ and $f(\mathbf{x}) < 0$ are labeled $+1$ and -1 respectively.

In Section 6.1, the application of the update scheme to problems with up to seven variables is presented. The evolution of the new convergence and error measures during the update are shown. Section 6.2 presents an example of SVM locking. In order to show the advantage of the new scheme in the removal of locking, a comparison to the adaptive sampling scheme without secondary sample evaluation is provided for this example.

The following notation will be used to present the results:

- $N_{initial}$ is the initial training set size.
- N_{total} is the total number of samples.
- $\epsilon_{initial}$ and ϵ_{final} are the test point-based errors associated with the initial and final SVM decision boundaries respectively.
- $E_{initial}$ and E_{final} are the errors associated with the initial and final SVM decision boundaries respectively, based on the comparison with the polynomial coefficients of the actual functions.

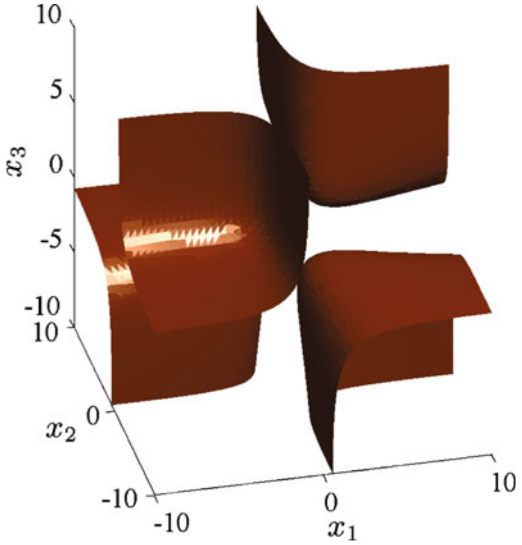

Fig. 6 Three-dimensional problem with disjoint regions. Actual decision boundary

6.1 Example 1: application to high dimensional problems

This section presents the application of the new update scheme to three analytical test functions of different

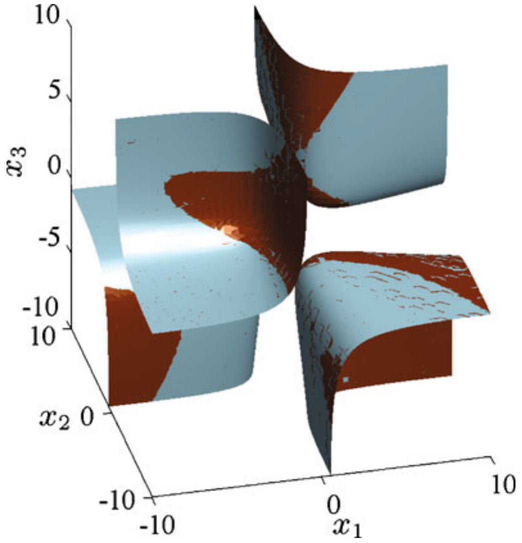

Fig. 7 Three-dimensional problem with disjoint regions. Updated SVM boundary (*light blue surface*) and the actual decision boundary (*dark brown surface*)

dimensionality that are derived from the same general equation. The functions presented consist of three, five and seven variables, and represent non-convex and disjoint regions. The general equation written as a function of the dimensionality d is:

$$f(\mathbf{x}) = \sum_{i=1}^{d} (x_i + 2\beta)^2 - 3 \sum_{j=1}^{d-2} \prod_{l=j}^{j+2} x_l + 1 = 0$$

$$\begin{aligned} \beta &= -1 \qquad mod(i,3) = 1 \\ \beta &= 0 \qquad mod(i,3) = 2 \\ \beta &= 1 \qquad mod(i,3) = 0 \end{aligned} \tag{14}$$

For example, the decision function in a three-dimensional case (15) is obtained by substituting $d = 3$ in the general equation. The actual boundary (decision function) for the three-dimensional case is plotted in Fig. 6). It forms several disjoint regions in the space.

$$f(\mathbf{x}) = (x_1 - 2)^2 + x_2^2 + (x_3 + 2)^2 - 3x_1x_2x_3 + 1 = 0 \tag{15}$$

The polynomial kernel is used to construct the SVM boundary in each of the examples. The degree of the polynomial is automatically selected as explained in Section 5.3.

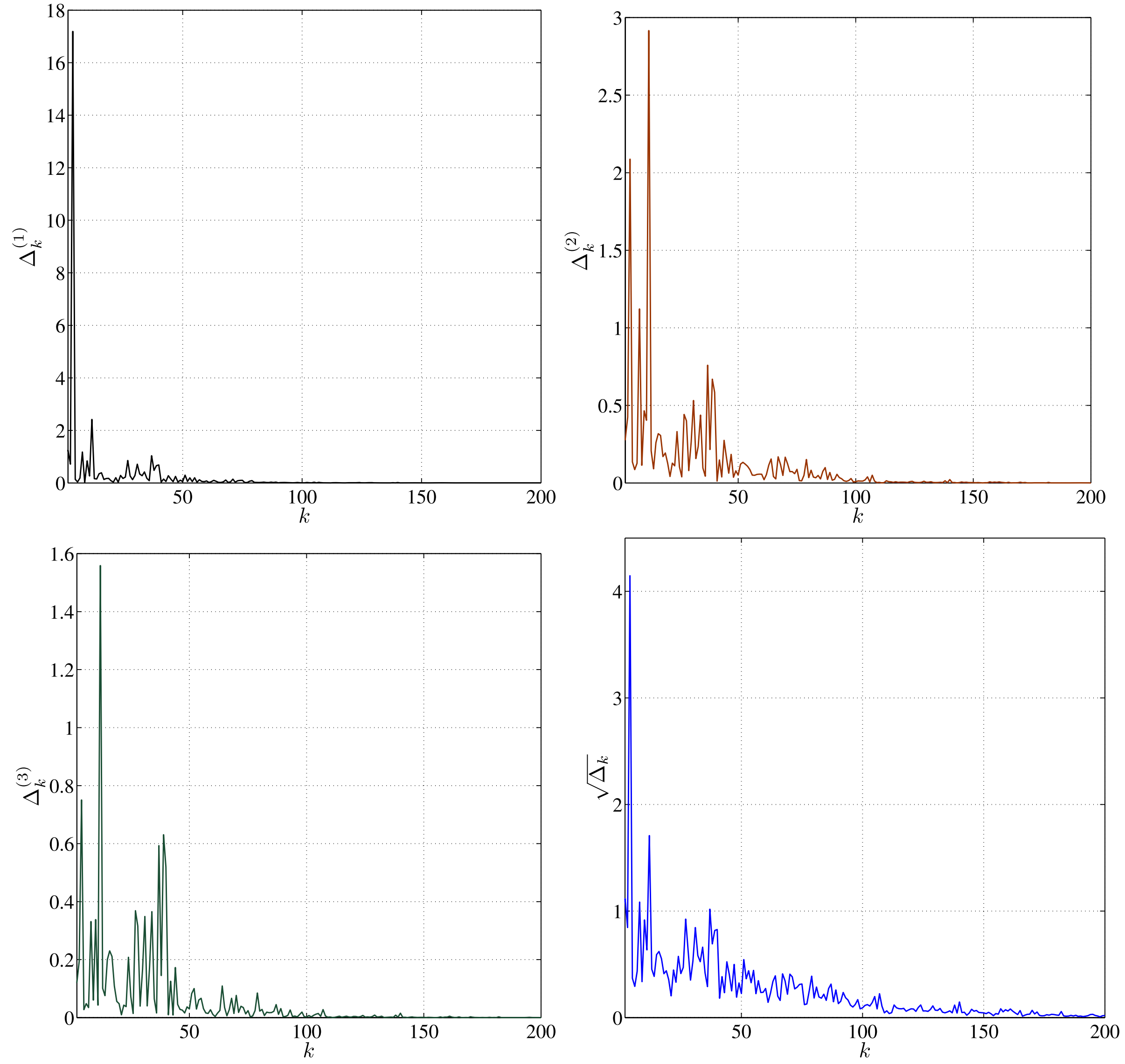

Fig. 8 Three-dimensional problem. The *bottom right figure* shows the square root of the convergence measure. The other figures show the variation of the coefficients corresponding to polynomial degrees 1, 2, 3

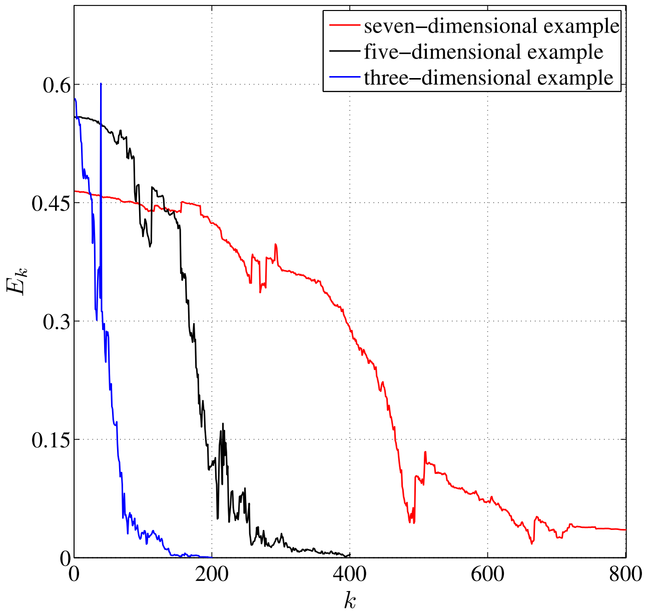

Fig. 9 Comparison of the errors associated with the three, five and seven-dimensional examples

As evident from (14), the actual decision functions are polynomials of degree 3. It is observed that for all the examples the algorithm automatically selects a polynomial kernel of degree 3 to construct the SVM boundary. Starting from relatively small CVT DOEs, the update algorithm is run up to a fixed number of iterations for each of the examples to study the evolution of the error and convergence properties of the algorithm. No actual convergence threshold is set for these problems. The initial and final values of the error measure ϵ_k are calculated using 10^7 test points for all the examples. For the optimization problems in (4) and (6), a convergence criterion of 10^{-3} was used on the objective function and the variables.

The results of the update for all three examples are listed in Table 1. The final SVM boundary for the three-dimensional case is plotted in Fig. 7. The convergence plot for the three-dimensional example is depicted in Fig. 8. The square root of the convergence measure Δ_k (10) is used for better readability of the plot by compressing the difference between the largest and the smallest values. The quantities $\Delta_k^{(1)}$, $\Delta_k^{(2)}$ and $\Delta_k^{(3)}$ (11) are also shown. The plot for $\Delta_k^{(1)}$ has a large peak in the beginning; however, being associated with the linear terms, this may not correspond to the largest change in the SVM. At the end of the update all the quantities ($\Delta_k^{(1)}$, $\Delta_k^{(2)}$ and $\Delta_k^{(3)}$) converge to zero. The errors (E_k) for the three examples are plotted together in Fig. 9. The initial and final values of the error measure ϵ_k are also provided in Table 1. The final error ϵ_{final}, which measures the discrepancy between the approximated and the actual boundary based a large number of test samples is lower than 0.1% even for the seven-dimensional example. Similarly, the error E_{final}, based on the polynomial coefficients, is also low. It must be emphasized that the latter measure, although less intuitive than ϵ_{final}, allows one to quantify the error in higher dimensional spaces.

6.2 Example 2: comparison of the update schemes with and without secondary sample evaluation

In order to depict the importance of evaluating secondary samples, which is a major addition in the proposed new update scheme, a two-dimensional analytical test example is presented. The equation of the actual decision boundary is:

$$f(\mathbf{x}) = x_2 - 2\sin(x_1) - 5 = 0 \qquad (16)$$

The initial SVM boundary is constructed using 20 CVT samples. It is then updated using the new proposed scheme. The update is run up to 50 iterations and the final SVM boundary is constructed with a polynomial kernel of degree 4. In order to demonstrate the effect of secondary sample evaluations, the results are compared to the SVM boundary obtained after 50 iterations using primary samples only. The final SVM boundaries using the two schemes are plotted in Fig. 10. The comparison of the evolution of the error measure ϵ_k, with and without secondary samples, is shown in Fig. 11. The final decision boundary using the new scheme is close to the actual boundary whereas the scheme without secondary sample evaluation displays the locking phenomenon in some localized regions.

Table 1 Number of samples and corresponding errors for the three examples

d	$N_{initial}$	$E_{initial}$ (%)	$\epsilon_{initial}$(%)	$Iterations$	N_{total}	E_{final} (%)	ϵ_{final}(%)
3	40	58.25	9.4	200	640	0.01	1.9×10^{-3}
5	160	55.93	14.1	400	1, 360	0.47	8.5×10^{-3}
7	640	46.44	8.6	800	3, 040	3.54	8.9×10^{-2}

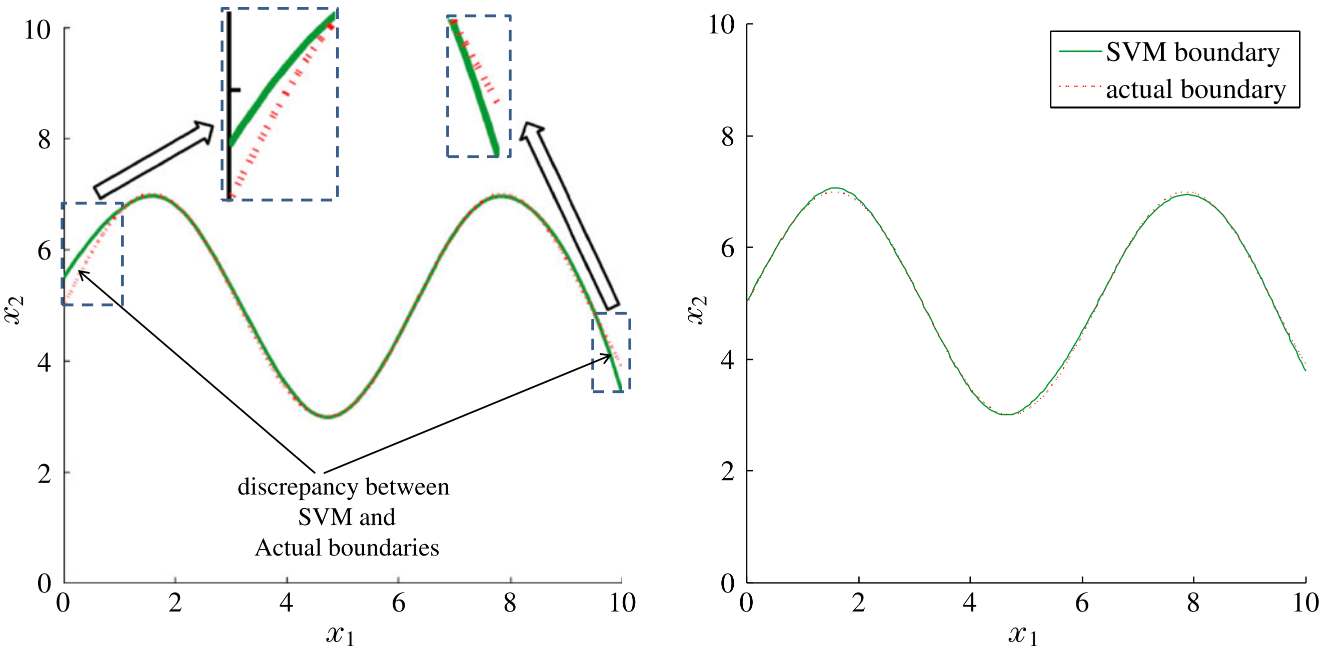

Fig. 10 Comparison of update schemes with (new scheme) and without (previous scheme) secondary samples. The regions where the SVM boundary using the old scheme differs from the actual boundary are circled in the left figure. The boundary using the new scheme (right) is very close to the actual boundary

7 Discussion

This section presents a discussion on some of the features of the SVM update. The effects of the new proposed method on the update, as well as some possible improvements are discussed.

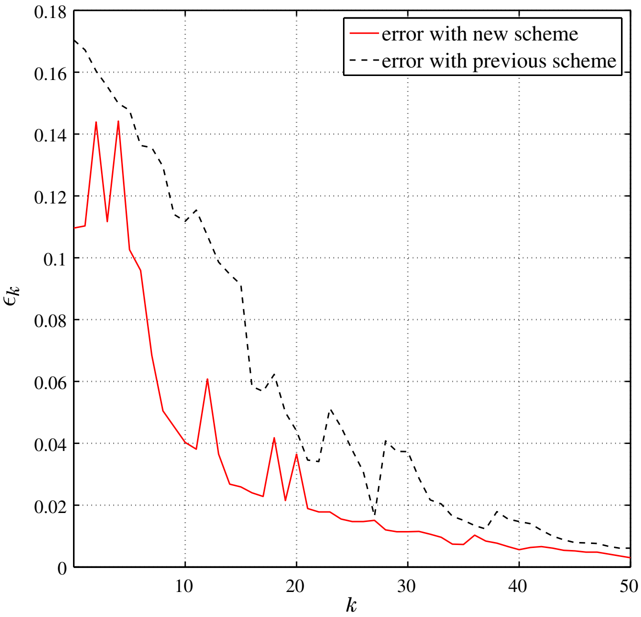

Fig. 11 Comparison of the evolution of error measure ϵ_k using the update schemes with (new scheme) and without (previous scheme) (Basudhar and Missoum 2008) secondary samples

- *Effect of the new sampling scheme on SVM locking and convergence of the update:* One of the most important contributions of this work is the identification of the "SVM locking" phenomenon as well as the development of a remedial solution. The proposed solution consists of using a "secondary" sample that leads to a more uniform distribution of samples in the vicinity of the locking. The locking phenomenon and the effect of the secondary samples are depicted in Figs. 3 and 5, as well as in an example in Section 6.2. It is noteworthy that although the term "SVM locking" may suggest that it entirely "stops" the SVM update, in reality, the update is believed to be convergent even without the locking removal step (secondary sample evaluation). However this would require a large number of samples, which would defeat one of the main purposes of the adaptive sampling scheme. The use of secondary samples enables one to reduce the number of necessary samples by efficiently reducing the local locking phenomena whose removal would otherwise require many function calls.

  Another noteworthy feature of the locking phenomenon stems from the fact that it is a local phenomenon. For this reason, there might not always be a clear difference between the global convergence rates of the proposed scheme and the adaptive sampling scheme without secondary samples. However, these local errors in the SVM boundary construction might have a significant influence on the optimum solution or the probability of failure calculated using the SVM boundary. Therefore, it is important to remove the locking using secondary samples. Also, it is expected that the locking phenomenon may have a greater influence on the global convergence rate in higher dimensions. This needs a detailed study in the future.

- *Optimization of the sampling sequence:* Although the proposed sampling scheme has some clear advantages in the removal of the locking, there is a scope for further improvement of the approach. The frequency of evaluating secondary samples is not optimized in this work; there is no scheme to detect whether or not a secondary sample is required. Therefore, secondary samples are selected systematically (one for every two primary samples) in regions that are most likely to require a secondary sample. Such regions are identified as the ones where data from one class is sparse in the vicinity of the boundary. A scheme to detect whether a secondary sample needs to be evaluated may be useful. Such a scheme may be devised based on a critical distance from existing samples. However, ways to define the critical distance need to be studied.
- *Choice of the kernel:* As mentioned in Section 5.4, the polynomial kernel used in this paper allows for a rigorous convergence measure based on the polynomial coefficients. Unlike the previous scheme (Basudhar and Missoum 2008), this does not require a large number of "convergence" points. However, the polynomial kernel is not necessarily superior to other kernels, such as the Gaussian kernel, in terms of the number of evaluations. A similar convergence criterion may be used with a Gaussian kernel by expanding the Gaussian kernel in order to compare polynomial coefficients. However, the number of terms in the expansion of the Gaussian kernel may be crucial and needs to be studied. In terms of the methodology to select new samples, the update will remain the same irrespective of the kernel.

8 Concluding remarks

8.1 Summary

A new adaptive sampling scheme for explicit design space decomposition with SVM decision boundaries has been developed. The ability of the method to accurately reconstruct analytical functions has been demonstrated for problems up to seven dimensions. The proposed algorithm consists of a set of improvements to a previous adaptive sampling methodology by the authors. Specifically, the sampling strategy has been revised and a more rigorous convergence measure has been developed. The new sampling scheme helps to remove the phenomenon of SVM locking. The results from the application of the approach to highly nonlinear examples of up to seven variables are promising. The examples consist of decision boundaries that form multiple disjoint regions in the space.

8.2 Future work

Although this paper introduces some major improvements to the previous method (Basudhar and Missoum 2008), it could benefit from some relatively minor incremental changes as mentioned in the discussion section. Improvements to further reduce the number of samples are being considered. Specifically, a scheme to detect whether a secondary sample needs to be evaluated may be useful. Also, the polynomial kernel has been used in this work as it provides a rigorous convergence criterion based on the polynomial coefficients. In the future, the method will be generalized by enabling the use of the polynomial coefficient based convergence criterion for the Gaussian kernel, as explained in the discussion section. In addition to the modifications to the methodology, the approach will also be applied to define decision boundaries for the reliability analysis and optimization of complex systems.

Acknowledgement The support of the National Science Foundation (award CMMI-0800117) is gratefully acknowledged.

References

Basudhar A, Missoum S (2007) Parallel update of failure domain boundaries constructed using support vector machines. In: 7th world congress on structural and multidisciplinary optimization. Seoul, Korea

Basudhar A, Missoum S (2008) Adaptive explicit decision functions for probabilistic design and optimization using support vector machines. Comput Struct 86(19–20):1904–1917

Basudhar A, Missoum S (2009a) A sampling-based approach for probabilistic design with random fields. Comput Methods Appl Mech Eng 198(47–48):3647–3655

Basudhar A, Missoum S (2009b) Local update of support vector machine decision boundaries. In: 50th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference. Palm Springs, California

Basudhar A, Missoum S, Harrison Sanchez A (2008) Limit state function identification using support vector machines for discontinuous responses and disjoint failure domains. Probab Eng Mech 23(1):1–11

Beachkofski BK, Grandhi R (2002) Improved distributed hypercube sampling. In: Proceedings of the 43rd AIAA/ASME/ASCE/AHS/

ASC structures, dynamics and materials conference. Paper AIAA-2002-1274, Denver, Colorado, USA

Bichon BJ, Eldred MS, Swiler LP, Mahadevan S, McFarland JM (2007) Multimodal reliability assessment for complex engineering applications using efficient global optimization. In: Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC structures, dynamics and materials conference. Paper AIAA-2007-1946, Honolulu, Hawaii

Bichon BJ, Mahadevan S, Eldred MS (2009) Reliability-based design optimization using efficient global reliability assessment. In: Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC structures, dynamics and materials conference. Paper AIAA-2009-2264, Palm Springs, California

Cawley GC, Talbot NLC (2003) Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers. Pattern Recogn 36(11):2585–2592

Gunn SR (1998) Support vector machines for classification and regression. Technical Report ISIS-1-98, Department of Electronics and Computer Science, University of Southampton

Helton JC (1993) Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal. Reliab Eng Syst Saf 42(2–3):327–367

Hurtado JE (2004) An examination of methods for approximating implicit limit state functions from the viewpoint of statistical learning theory. Struct Saf 26:271–293

Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13(4):455–492

Ma N (2001) Complete multinomial expansions. Appl Math Comput 124(3):365–370

Missoum S, Ramu P, Haftka RT (2007) A convex hull approach for the reliability-based design of nonlinear transient dynamic problems. Comput Methods Appl Mechanic Eng 196(29):2895–2906

Mourelatos ZP, Kuczera RC, Latcha M (2006) An efficient monte carlo reliability analysis using global and local metamodels. In: 11th AIAA/ISSMO multidisciplinary analysis and optimization conference, Pourtsmouth, Virginia, USA

Myers RH, Montgomery DC (2002) Response surface methodology, 2nd edn. Wiley, Hoboken

Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK (2008) Multiple surrogate modeling for axial compressor blade shape optimization. J Propuls Power 24(2):302–310

Romero VJ, Burkardt JV, Gunzburger MD, Peterson JS (2006) Comparison of pure and "latinized" centroidal voronoi tesselation against various other statistical sampling methods. J Reliab Eng Sys Saf 91:1266–1280

Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge

Simpson TW, Toropov VV, Balabanov VO, Viana FAC (2008) Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come – or not. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, Reston, VA

Tou JT, Gonzalez RC (1974) Pattern recognition principles. Addison-Wesley, Reading

Vapnik VN (1998) Statistical learning theory. Wiley, Hoboken

Wang GG, Shan S (2007) Review of metamodeling techniques in support of engineering design optimization. J Mech Des 129(4):370–380