Computers and Structures 86 (2008) 1904–1917

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Adaptive explicit decision functions for probabilistic design and optimization using support vector machines

Anirban Basudhar, Samy Missoum *

Aerospace and Mechanical Engineering Department, The University of Arizona, Tucson, AZ 85721, United States

ARTICLE INFO

Article history:
Received 20 August 2007
Accepted 27 February 2008
Available online 15 May 2008

Keywords:
Support vector machines
Explicit decision functions
Discontinuities
Disjoint failure regions
Optimization
Probabilistic design

ABSTRACT

This article presents a methodology to generate explicit decision functions using support vector machines (SVM). A decision function is defined as the boundary between two regions of a design space (e.g., an optimization constraint or a limit-state function in reliability). The SVM-based decision function, which is initially constructed based on a design of experiments, depends on the amount and quality of the training data used. For this reason, an adaptive sampling scheme that updates the decision function is proposed. An accurate approximated explicit decision functions is obtained with a reduced number of function evaluations. Three problems are presented to demonstrate the efficiency of the update scheme to explicitly reconstruct known analytical decision functions. The chosen functions are the boundaries of disjoint regions of the design space. A convergence criterion and error measure are proposed. The scheme is also applied to the definition of an explicit failure region boundary in the case of the buckling of a geometrically nonlinear arch.

1. Introduction

The simulation-based design of complex engineering applications is often associated with high computational costs, thus making design optimization and reliability assessment tedious. In order to reduce the computational burden, actual models are often replaced by surrogates such as response surfaces or metamodels [1]. These approximations are embedded within an optimization loop or are used to efficiently run Monte-Carlo simulations (MCS) [2]. Surrogates are typically built using the values of the system's responses for selected design configurations defined by a design of experiments (DOE) [3]. However, the accuracy of the approximation greatly depends on the amount and quality of training data used. It is well known that the filling of the design space with samples is limited to a few dimensions because of the so-called "curse of dimensionality": the number of samples needed increases exponentially with the problem dimensionality.

In the literature, some approaches have been proposed to reduce the number of samples by selectively choosing them. In [4], a reliability assessment method was proposed whereby additional points were generated in the vicinity of an implicitly defined limit-state function. For this purpose, a sampling guidance function was defined based on the difference between the value of the performance function at a point (approximated by a metamodel) and the allowable performance value. A set of new samples with high guidance function values were selected from a uniform grid. Among these, the point having the maximum probability of failure was chosen as the new training sample. Another approach used an expected improvement function (EIF) to select the location for new training samples [5,6] in order to update and refine a Kriging response approximation.

This article introduces a new adaptive sampling scheme which reduces the number of function evaluations. However, instead of approximating responses and using implicitly defined optimization constraints or limit-state functions, the proposed approach constructs explicit approximation of these boundaries with respect to the design variables [7]. That is, the design space is explicitly decomposed into feasible and infeasible regions (or failure and safe regions if reliability is considered). For the sake of clarity, we will refer to constraints and limit-state functions as decision functions for the remainder of this article.

The approach, which does not approximate responses, has the advantage of avoiding the difficulties due to discontinuous responses often encountered in nonlinear problems. In simulation-based design, discontinuities present a serious problem for optimization or probabilistic techniques because system responses are usually assumed continuous. In optimization, this restricts any traditional gradient-based method or response surface technique. Discontinuities can be detected by data mining techniques such as clustering [8], which automatically identifies groups of similar responses. It is then possible to map these clusters to specific regions of the design space.

* Corresponding author. Tel.: +1 520 626 5226; fax: +1 520 621 8191.
E-mail address: smissoum@email.arizona.edu (S. Missoum).

0045-7949/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compstruc.2008.02.008

When considering reliability, discontinuities might also hamper the use of approximation methods such as first and second order reliability methods (FORM and SORM) [9], advanced mean value (AMV) [10], or Monte-Carlo simulations with response surfaces. In addition to discontinuities, nonlinear problems are characterized by disjoint failure regions, thus further limiting the use of classical approaches to assess probabilities of failure. These disjoint regions are often associated with distinct system behaviors, a phenomenon that is found in structural impact problems (e.g., vehicle crash) [11]. However, by explicitly defining the boundaries of the possibly disjoint failure domain, the calculation of failure probabilities through Monte-Carlo simulations is made efficient, as the verification of the state of a sample (failed or safe) is straightforward and computationally efficient.

Several attempts have been made to explicitly decompose the design space in the case of nonlinear transient dynamic problems. Hyperplanes and ellipsoids, defined explicitly with respect to design variables, were first used in the case of a tube impacting a rigid wall [12]. These decision functions split the design space into two regions corresponding to crushing and global buckling behaviors. The decision functions were then used to optimize the tube so as to avoid buckling while taking uncertainties into account. The boundaries of the failure region (global buckling) were later defined with a convex hull which led to a more accurate and less conservative failure domain [13].

However, the tools used to create decision functions were not satisfactory as they were limited to a single convex set, and therefore did not address the issue of non-convex disjoint failure domains. The approach was generalized by constructing the boundaries of specific regions of the design space using support vector machines (SVM) [14–16]. SVM is a powerful classification tool that enables the construction of linear or nonlinear optimal decision functions between classes in a multi-dimensional space. The decision functions can be non-convex and form several disjoint subsets.

The explicit design space decomposition is made possible by first studying the responses with a DOE. In order to distribute the samples uniformly over the design space, techniques such as improved distributed hypercube sampling (IHS) [17] or Latinized centroidal Voronoi tesselation (LCVT) [18] can be used. The responses, obtained for each DOE sample, are then categorized into "acceptable" or not (e.g., safe or failed). This classification enables the use of SVM to construct explicit decision functions.

However, the number of training samples needed for the construction of an accurate decision function depends on the complexity of the function and the number of dimensions of the problem. In general, it is difficult to predict the required training set size. Unless a very large DOE is generated, which is not practical for most problems, the first decision function might not be accurate and needs to be updated.

In this paper, an algorithm to update the initial decision function starting from a small training set size is described. It is an adaptive sampling strategy based on the selection of points that lie on the SVM decision function [19]. It can be shown that such samples are bound to modify the decision function and therefore constitute a natural element of the update approach. These samples are efficiently found by a global optimization technique such as a genetic algorithm (GA). A stopping criterion is proposed which dictates the number of training samples used to construct the decision function. The approach is applied to the reconstruction of three analytical problems with two, three and four variables. These test decision functions form the boundaries of non-convex and disjoint failure regions. An error metric is introduced to quantify the error between the approximated decision function and the actual analytical function. In addition, the accuracy of the updated decision function is compared to that of a function trained with an LCVT distribution using the same number of samples. In addition to the analytical functions, the methodology is applied to an arch structure with three random parameters. Due to buckling, the response (largest displacement) is discontinuous. These discontinuities are identified using a clustering technique which provides the basic classification for the construction of the SVM decision function.

2. Support vector machines

SVM is a machine learning technique that is becoming increasingly popular and has widespread applications in classification and pattern recognition [14,15]. A variation of SVM is used as a regression tool and is referred to as support vector regression (SVR) [20]. The main feature of SVM lies in its ability to define complex decision functions that optimally separate two classes of data samples. The purpose of this section is to provide the reader with an overview of the SVM algorithm.

Consider a set of N training samples $\mathbf{x}_i$ in a d-dimensional space. Each sample is associated with one of two classes characterized by a value $y_i = \pm 1$. The SVM algorithm finds the boundary (decision function) that optimally separates the training data into the two classes. The basic SVM theory is presented through a detailed explanation in the case of a linearly separable data set. It is then extended to the case where the data is not linearly separable.

2.1. Linear decision function

In the SVM theory, the linear decision function lies half way between two hyperplanes that separate the two classes of data. This pair of hyperplanes, referred to as "support hyperplanes", is required to pass at least through one of the training samples of each class (*support vectors*) while no sample can be found within the margin (Fig. 1). For separable data, there are an infinity of possible decision functions. In order to find the "optimal" decision function, the basic idea is to maximize the "margin" that separates the support hyperplanes. One of the support hyperplanes consists of those points that satisfy:

$$\mathbf{w} \cdot \mathbf{x} + b = +1 \tag{1}$$

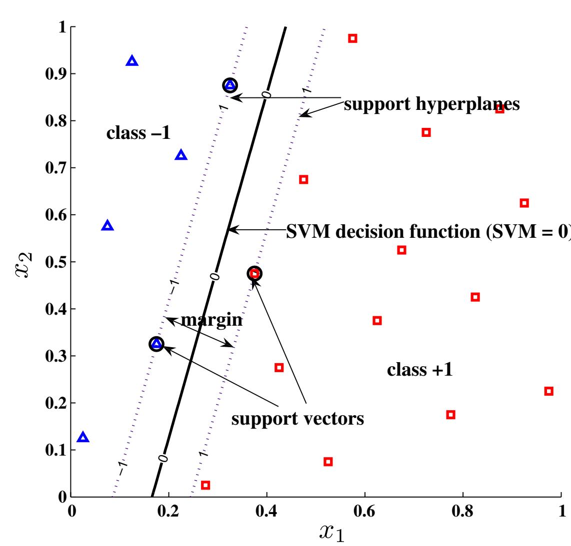

Fig. 1. Linear decision function separating class +1 (red squares) from class −1 (blue triangles). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The other hyperplane contains the points that follow:

$$\mathbf{w} \cdot \mathbf{x} + b = -1 \tag{2}$$

where $\mathbf{x}$ is the position vector of a point in space, $\mathbf{w}$ is the vector of hyperplane coefficients and b is the bias. All the points of the class $y = +1$ lead to a positive value of SVM and all the points in the class $y = -1$ are "negative". Eqs. (1) and (2), and the constraint that no sample can lie between the two aforementioned hyperplanes, can be combined in a single global constraint defined as follows:

$$y_i(\mathbf{w} \cdot \mathbf{x} + b) - 1 \geqslant 0 \tag{3}$$

The perpendicular distance between the two support hyperplanes is $\frac{2}{\|\mathbf{w}\|}$. Therefore, determining the support hyperplanes (i.e., solving for $\mathbf{w}$ and b) reduces to the following optimization problem

$$\begin{aligned} &\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 \\ &y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geqslant 0 \qquad 1 \leqslant i \leqslant N \end{aligned} \tag{4}$$

This is a quadratic programming (QP) problem since the objective function is quadratic, and the constraints are linear. Problem (4) is convex and can be solved efficiently with available optimization packages. As a result, the optimal $\mathbf{w}$, b, and the Lagrange multipliers λ_i at the optimum are obtained. From this, the classification of any test point $\mathbf{x}$ is obtained by the sign of the following function:

$$s = b + \sum_{i=1}^{N} \lambda_i \, y_i \, \mathbf{x}_i \cdot \mathbf{x} \tag{5}$$

Note that, following the Kuhn and Tucker conditions, only the Lagrange multipliers associated with the support vectors will be strictly positive while the other ones will be equal to zero. In general, the number of support vectors is a small fraction of the total number of training samples. Eq. (5) can be rewritten with respect to the number of support vectors *NSV*:

$$s = b + \sum_{i=1}^{NSV} \lambda_i \, y_i \, \mathbf{x}_i \cdot \mathbf{x} \tag{6}$$

In the case where the data is not linearly separable, the optimization problem (4) will be infeasible. The inequality constraints are then relaxed by the introduction of non-negative slack variables ξ_i which are minimized through a penalized objective function. The relaxed optimization problem is

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ &y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \geqslant -\xi_i \qquad 1 \leqslant i \leqslant N \end{aligned} \tag{7}$$

The coefficient C is referred to as the misclassification cost. In the dual formulation of Problem (7), C becomes the upper bound for all the Lagrange multipliers.

2.2. Nonlinear decision function

SVM can be extended to the case of nonlinear decision functions by projecting the original set of variables to a higher dimensional space referred to as the feature space. In this n dimensional feature space, the new components of a point $\mathbf{x}$ are given by $(\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_n(\mathbf{x}))$ where ϕ_i are the features. The remarkable feature of SVM is that the nonlinear decision function is obtained by formulating the linear classification problem in the feature space. The classification is then obtained by the sign of

$$s = b + \sum_{i=1}^{NSV} \lambda_i y_i < \mathbf{\Phi}(\mathbf{x_i}), \mathbf{\Phi}(\mathbf{x}) > \tag{8}$$

where $\mathbf{\Phi} = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_n(\mathbf{x}))$ and $<,>$ is the inner product.

The inner product in Eq. (8) forms a kernel K, so that the decision function is written:

$$s = b + \sum_{i=1}^{NSV} \lambda_i \, y_i \, K(\mathbf{x}_i, \mathbf{x}) \tag{9}$$

2.3. Types of kernels

The two most commonly used kernels functions are the polynomial and the Gaussian kernels. Some other kernels that may be used are multi-layer perceptions, Fourier series, and splines [21]. The Gaussian kernel used in this paper is defined as

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}\right) \tag{10}$$

where σ is the width factor of the Gaussian kernel. An example of classification using a Gaussian kernel is provided in Fig. 2.

2.4. General features of SVM

SVM has several features which make it a very powerful tool for pattern recognition and classification. These features are a also useful in probabilistic design and optimization. Some features of SVM are:

1. SVM is multi-dimensional: SVM is capable of classifying data in a multi-dimensional space. In Fig. 3 a Gaussian kernel in three dimensions is used to optimally define the boundary between class +1 (red squares) and class −1 (blue triangles).
2. Optimal decomposition: There can be several ways to separate two classes of data. However, SVM decomposes the design space by an optimal separating function which maximizes the margin between the classes.
3. Separation of disjoint regions: SVM is capable of identifying disjoint regions. Hence, it can be applied to problems for which the decision function forms the boundaries of several disjoint regions in the design space.

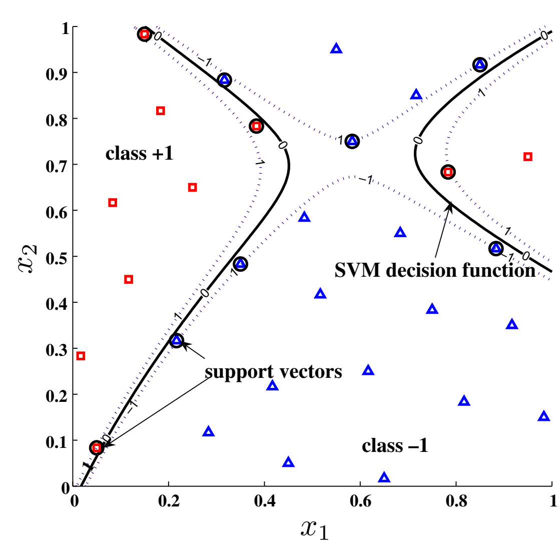

Fig. 2. Two-dimensional Gaussian kernel separating the two classes shown by blue triangles and red squares. The zero value iso-contour represents the optimal decision function and the support vectors are shown with circles. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

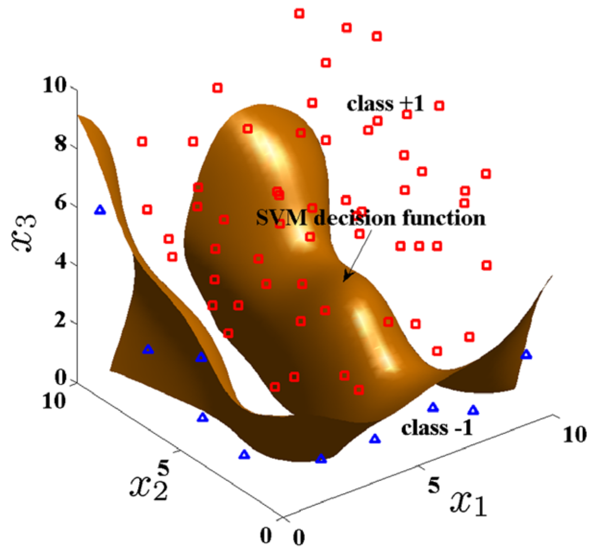

Fig. 3. Three-dimensional Gaussian kernel separating the two classes shown by blue triangles and red squares. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3. Methodology for the identification of explicit decision functions

The methodology used for the identification of explicit boundaries with SVM is presented in this section. The first attempt to use SVM for design space decomposition was presented in [16]. The focus of the paper was the use of explicit design space decomposition to handle the probabilistic design of problems with discontinuities. However, the number of training samples required for an accurate estimate of the decision function was quite large. Further, there was no specific criterion to decide the number of training samples that are needed, as it depends on the specific problem at hand.

In order to address these issues, an update scheme was derived. The problem of using training samples in the most efficient way can be seen as an approach to update the decision function as well as a way to improve the DOE for a specific problem. To achieve this objective, this article proposes an active learning sample selection technique that iteratively uses the information from the previously constructed SVM decision function.

The first step consists of performing an initial DOE [22] to sample the design space. The responses of the system for the DOE samples are then evaluated and classified into distinct classes that correspond to feasible or infeasible system behaviors. These classified design configurations are then used as training samples for the SVM algorithm. For practical purposes, the initial DOE size is typically maintained rather small, and therefore, the initial prediction of the decision function may be inaccurate. The update algorithm is then used to refine it. The three major steps of the approach are described in the sequel and summarized in Fig. 4.

3.1. Design of experiments – LCVT

There exist several DOE techniques such as Latin hypercube sampling (LHS) [23], D-optimal sampling [3], improved and optimum Latin hypercube sampling (IHS and OLHS) [17,24]. In our approach, the initial training sample set is generated using LCVT [18]. LCVT is chosen for the training sample distribution because it tends to provide a uniform distribution of information within the design space while retaining the characteristics of a Latin hypercube. Fig. 5 provides examples of both LCVT sampling and LHS. The sampling obtained using LCVT is seen to be more uniformly distributed, although it has slightly higher discrepancy.

3.2. Estimation and classification of responses

After generating the LCVT DOE, response values at these sample points are evaluated and classified. In the general case, these responses might be obtained by a simulation code. They are then classified by comparing them to a "threshold" value or by the use of clustering. The threshold value is the traditional allowable response value used to define feasibility in optimization or failure in reliability. However, in the case of discontinuous responses, such a threshold might not be known a priori and cluster identification techniques such as K-means [25] or Hierarchical clustering [8,26] are needed. The classification of responses into two distinct classes (e.g., safe or failed) provides the information needed by the SVM algorithm to generate the decision function. Fig. 6 shows two cases in which the samples are classified using a threshold value and clustering.

3.3. Definition of an explicit decision function – update algorithm

Following the selection of training samples with a DOE, and the classification of response values, SVM is used to generate an explicit decision function. The construction of the initial approximated decision function is then followed by the update. The basic idea is to choose a new sample point that is likely to modify the predicted decision function when added to the training set. The following two criteria help to achieve that objective:

- A new training sample is selected such that it has the highest probability of being misclassified by the SVM decision function. Such points are clearly located on the decision function itself (i.e., SVM = 0). In addition, the new training sample selected on the decision function lies within the SVM margin which, by construction, does not include any sample. Therefore, the decision function is bound to be modified.
- A new training sample should not be near existing sample points in order to avoid redundant information and useless function evaluations. For this, a minimum distance between samples is enforced as a function of the hypervolume of the design space, the problem dimensionality, and the number of training samples.

The update algorithm is based on the following steps:

Step 1: *Choice of training samples on the decision function* – In the first step of the algorithm, a new training sample is selected according to the two aforementioned criteria. The corresponding problem is:

$$\begin{aligned} &\min_{\mathbf{x}} \quad \left| b + \sum_{i=1}^{NSV} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) \right| \\ &\quad l_x \geqslant \alpha \left(\frac{V}{N}\right)^{\frac{1}{d}} \end{aligned} \tag{11}$$

where l_x is the distance of a point from the nearest existing training sample, and the right hand side of the inequality represents the minimum allowable distance. V is the hypervolume of the d-dimensional space and $0 < \alpha \leqslant 1$. This search is a global optimization problem that is solved by a genetic algorithm (GA). Fig. 7 shows the effect of adding a new training sample on the decision function. After the new sample is selected, the SVM decision function is updated.

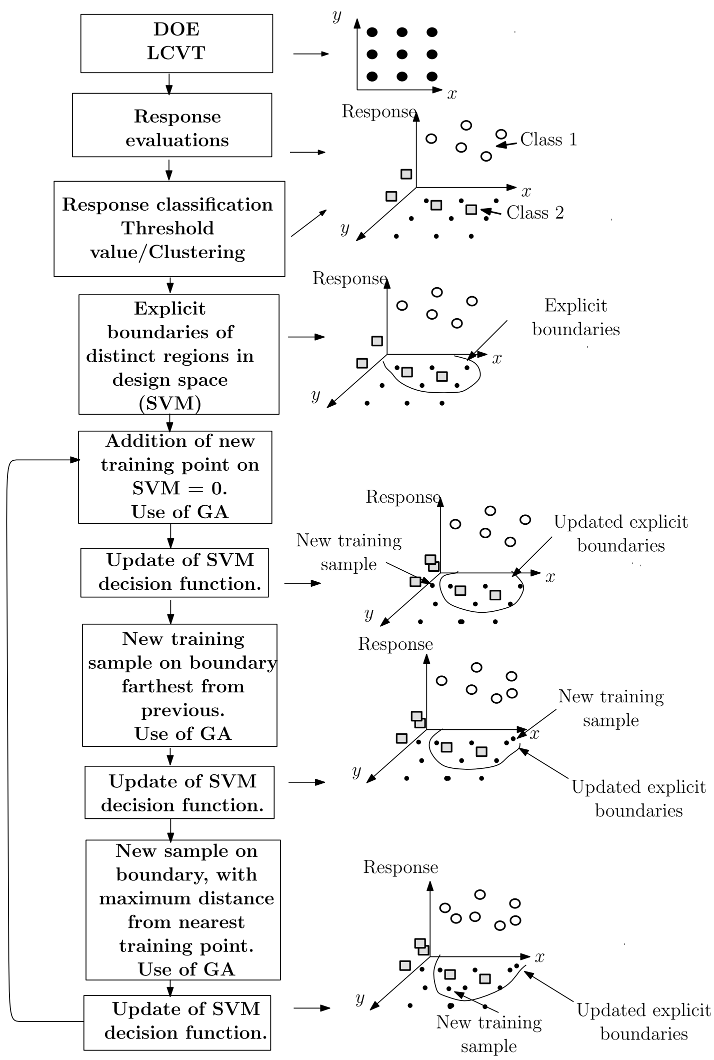

Fig. 4. Explicit identification of boundaries with the application of the update algorithm.

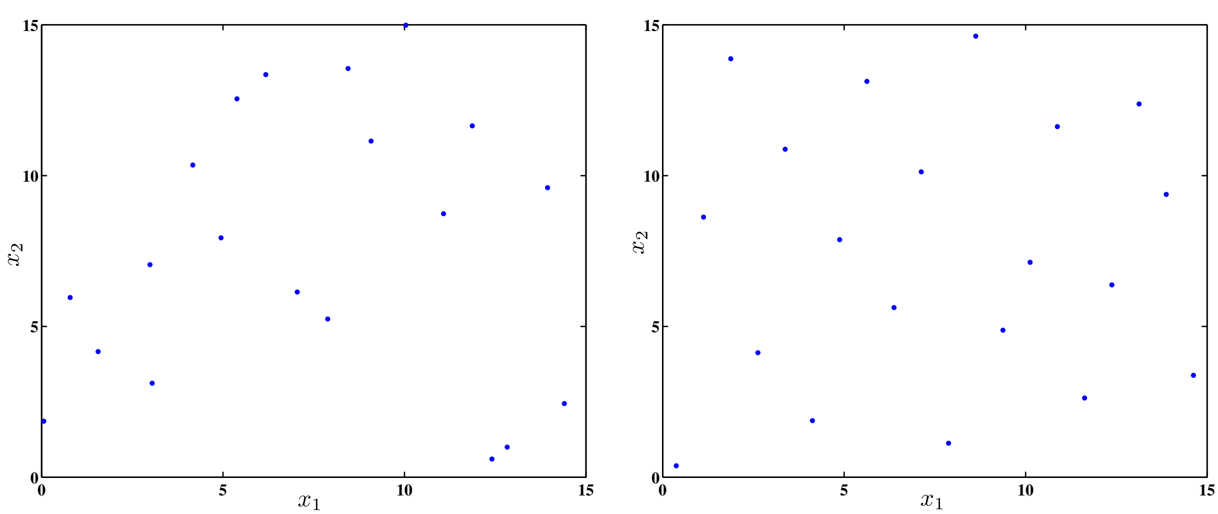

Fig. 5. Comparison of the uniformity of two-dimensional sample distributions using LHS (left) and LCVT (right).

Step 2: *Sample on decision function at maximum distance from the previously added point* – The possibility of new training samples being chosen in a localized region of the design space needs to be avoided. In step 2, a GA is used to find a point farthest from the previously added training sample, while following the two aforementioned criteria

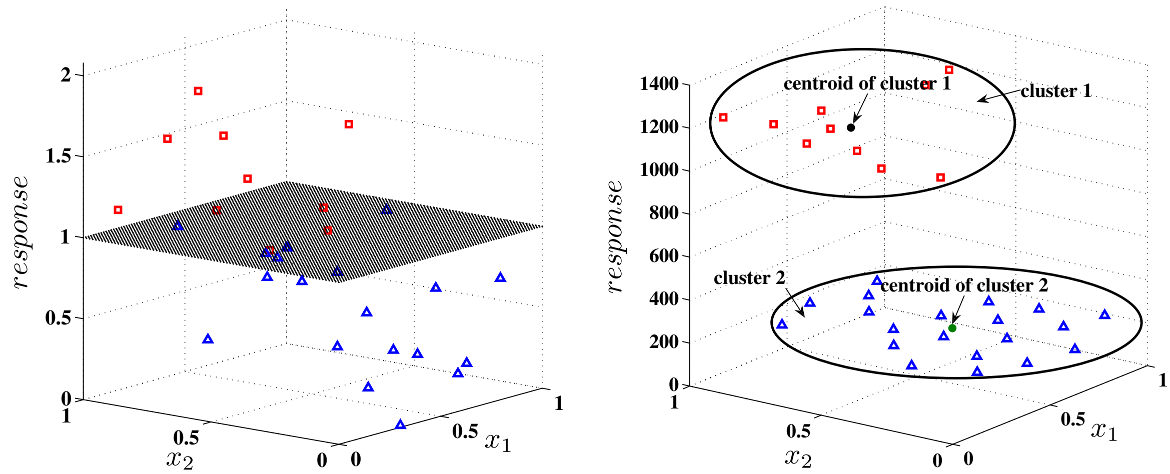

Fig. 6. Classification of responses by comparison to a known threshold value (left). Classification of discontinuous responses using clustering (right).

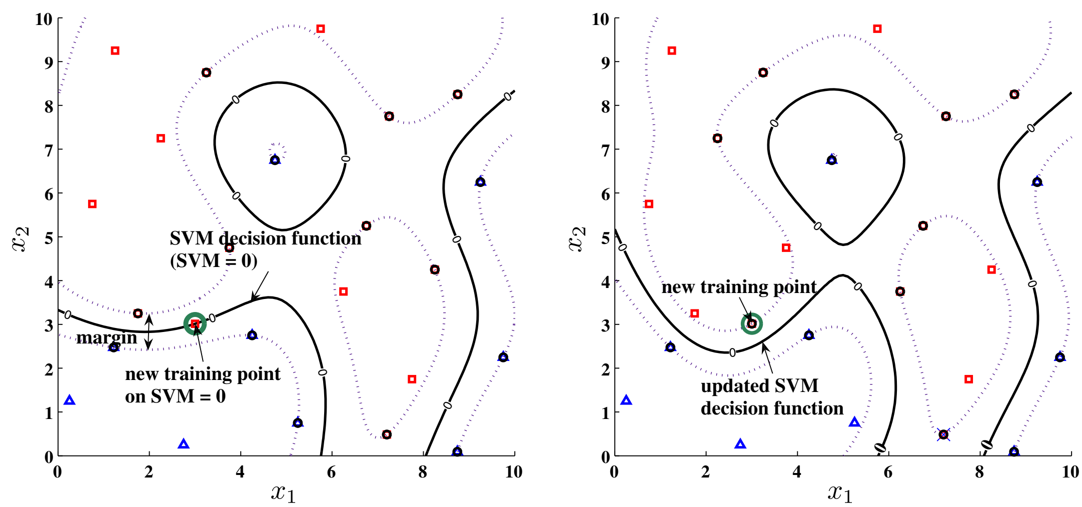

Fig. 7. Addition of a new training sample on SVM = 0. The left figure shows the initial decision function which is updated in the right hand side figure by adding a new sample on SVM = 0. The dotted curves represent the support functions (SVM = ±1).

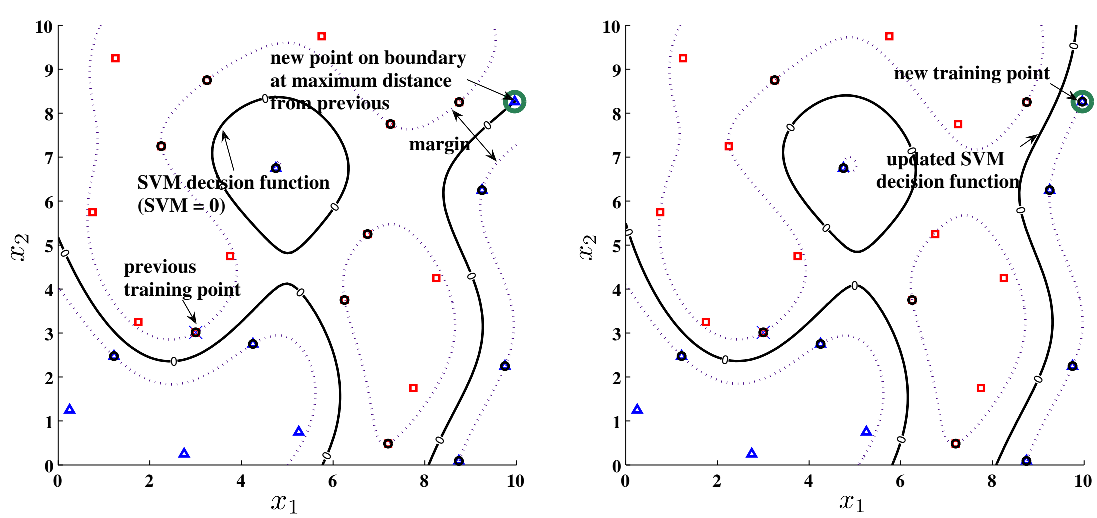

Fig. 8. Addition of a new training sample on the boundary that is farthest from the previously added training sample. The left figure shows the new training sample being added and the initial decision function. The updated function is shown on the right hand side figure. The dotted curves represent the support functions (SVM = ±1).

(Eq. (12)). After adding this new point to the training set, the SVM decision function is updated. This is illustrated in Fig. 8.The optimization problem, solved with a GA, is:

$$\begin{aligned} &\max_{\mathbf{x}} \ \|\mathbf{x}-\mathbf{x}_{\text{prev}}\| \\ &\quad l_x \geqslant \gamma\left(\frac{V}{N}\right)^{1/d} \\ &\quad \left|b+\sum_{i=1}^{NSV}\lambda_i y_i K(\mathbf{x}_i,\mathbf{x})\right| = 0 \end{aligned} \tag{12}$$

where $\mathbf{x}_{\text{prev}}$ is the previously added training sample, and γ is a coefficient less than 1.

Step 3: *Sample on decision function with maximum minimum distance from existing training samples* – In step 3, a GA is used to find a point on the decision function, which does not have any existing training sample in its neighborhood. For this purpose, the distance to the nearest existing training sample is maximized (Eq. (13)). The new point is included in the training set and the SVM decision function is reconstructed (Fig. 9).The optimization problem is

$$\begin{aligned} &\max_{\mathbf{x}} \ \|\mathbf{x}-\mathbf{x}_{\text{nearest}}\| \\ &\quad \left|b+\sum_{i=1}^{NSV}\lambda_i y_i K(\mathbf{x}_i,\mathbf{x})\right| = 0 \end{aligned} \tag{13}$$

where $\mathbf{x}_{\text{nearest}}$ is the nearest training sample from the current GA point being evaluated.The three steps in the update section are repeated until the stopping criterion is met.

3.3.1. Stopping criterion

In order to terminate the update algorithm, a stopping criterion is required. Because the actual explicit decision function is not known in general, the criterion is based on the variations of the approximated decision function. For this, a set of N_{conv} "convergence points" is generated using an LHS DOE. The fraction of convergence points for which there is a change of sign between two successive iterations is calculated. The number N_{conv} can be chosen to be quite high because the calculation of SVM values using Eq. (9) is inexpensive. For a d-dimensional space, N_{conv} is chosen as $100 \times 5^{\text{d}}$. Since the convergence points are generated using LHS, the generation of these samples is efficient. By choosing a large set of convergence points, an accurate estimate of the fraction can be achieved (Eq. (14)).

$$\Delta_k = \frac{\text{num}\left(|\text{sign}(s^i_{k-1}) - \text{sign}(s^i_k)| > 0\right)}{N_{\text{conv}}} \tag{14}$$

where Δ_k is the fraction of convergence points for which the sign of the SVM evaluation changes between iterations $k-1$ and k. s^i_{k-1} and s^i_k represent the SVM value of the i^{th} convergence point at iterations $k-1$ and k respectively. Change in the SVM decision function is very significant during early stages of the update and reduces gradually, as the quality of the approximation increases.

In order to implement a practical stopping criterion, the fraction of convergence points changing sign between successive iterations is fitted by an exponential curve

$$\widehat{\Delta}_k = Ae^{Bk} \tag{15}$$

where $\widehat{\Delta}_k$ represents the fitted values of Δ_k. A and B are the parameters of the exponential curve.

The value of $\widehat{\Delta}_k$ at the last iteration k_c is checked after each training sample is added. The slope of the curve is also calculated. For the update to stop, the value of the fitted curve should be less than a small positive number ϵ_1. Simultaneously, the absolute value of the slope of the curve at convergence should be lower than ϵ_2.

$$\begin{aligned} &Ae^{Bk_c} < \epsilon_1 \\ &-\epsilon_2 < BAe^{Bk_c} < 0 \end{aligned} \tag{16}$$

3.4. Error measure

The accuracy of the SVM decision function is judged by its fidelity to the actual decision function. In practical problems, an error metric is difficult or impossible to obtain. However, in the case of academic analytical test functions, an error measure can be obtained. For this purpose, a dense grid of N_{test} "test" points is generated over the whole space. The values of both the actual decision function and the SVM are calculated for each test point. Since the actual decision 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. A test point for which the sign of SVM does not match the sign provided by the actual function is considered misclassified. That is, the error ϵ is

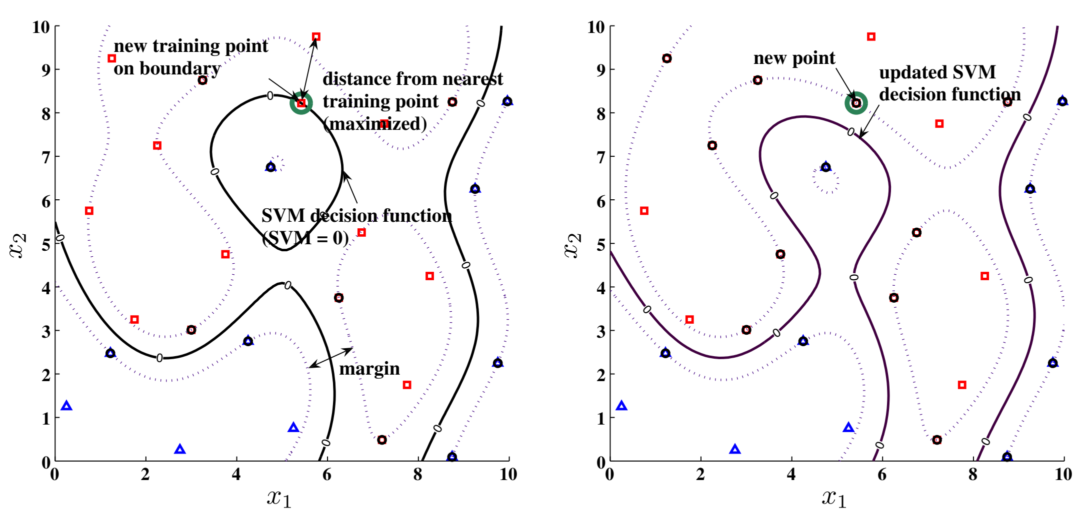

Fig. 9. Addition of a new training sample on the boundary with maximum minimum distance from existing training samples. The left figure shows the initial decision function. The updated function is shown on the right hand side figure. The dotted curves represent the support functions (SVM = ±1).

$$\epsilon = \frac{\mathrm{num}\left(\left(b + \sum_{i=1}^{NSV} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}_{\mathrm{test}})\right) y_{\mathrm{test}} \leqslant 0\right)}{N_{\mathrm{test}}} \quad (17)$$

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

4. Examples

Four test examples demonstrating the efficiency of the update methodology are used. Three of the problems consist of analytical decision functions representing non-convex and disjoint failure domains. Analytical functions allow one to verify if the proposed SVM update scheme has the ability to reproduce the decision functions. Problems with two, three, and four variables are studied. Also, the update scheme is applied to construct the explicit decision function in case of an arch structure having discontinuous response.

In all the problems, a Gaussian kernel with a width factor σ equal to 2.2 is used and the misclassification coefficient C is set to infinity to avoid misclassification. The value of σ depends on the ranges of the variables x_i and the complexity of the decision function. In the general case, an appropriate value of σ can be determined by minimizing the number of support vectors. The coefficients α and γ are equal to 0.2 and 0.5 respectively.

The following notation will be used in the result section:

- N_{initial} is the initial training set size.
- N_{total} is the total number of samples required at the end of the update.
- $\epsilon_{\mathrm{initial}}$ and $\epsilon_{\mathrm{final}}$ are the errors associated with the initial and final SVM decision functions respectively.
- ϵ_{LCVT} is the error associated with a decision function constructed with an LCVT sample distribution of N_{total} points.

The analytical decision functions are written in the form $f(x_k) = 0$, where x_k are the variables. In order to perform the SVM classification, the samples corresponding to $f(x_k) > 0$ and $f(x_k) < 0$ are labeled +1 and -1 respectively.

To better analyze the problems, studies have also been performed with respect to the initial training set size and the stopping criterion. In most cases, the stopping condition on ϵ_1 is the governing condition. Therefore, the study with respect to the stopping criterion is performed by varying ϵ_1 while the value of ϵ_2 is 5.0×10^{-4}.

4.1. Two-dimensional non-convex example with disjoint regions

For this problem, the decision function is defined by an analytical function of two variables x_1 and x_2.

$$f(x_1, x_2) = x_2 - |\tan(0.5x_1 + 2)| - 3 \quad (18)$$

The variables x_1 and x_2 are continuous and both belong to the interval [0,10]. As depicted in Fig. 11, the function, in dotted line, forms two disjoint regions.

4.1.1. Construction of the decision function

The initial decision function is constructed with 20 training samples generated using LCVT. The number of convergence points N_{conv} is 2500. The values of ϵ_1 and ϵ_2 for the stopping criterion are 4.0×10^{-3} and 5.0×10^{-4} respectively. For measuring the error, the fraction of misclassified test points is calculated as described in Section 3.4. The number of test points for the error measurement is N_{test} = 10,000.

The results are gathered in Table 1. The error in predicting the initial decision function using LCVT distribution is 8.74%, which reduces to 1.58% after the update. The total number of training samples needed is 75. In comparison, the decision function constructed with the same number of training samples generated by LCVT gives an error of 4.90%. Fig. 10 depicts the SVM decision function, in solid line, with initial and final training sets. For completeness, two other intermediate SVM decision functions, constructed with 40 and 60 training samples, are also shown. A decision function obtained by using a 75 initial LCVT samples is also shown in Fig. 11.

Table 1
Two-dimensional problem. Effect of the initial LCVT training set size. The errors are noted for SVM decision functions constructed using the update scheme and by using a static LCVT distribution with the same number of samples.

N_{initial}	$\epsilon_{\mathrm{initial}}$	N_{total}	$\epsilon_{\mathrm{final}}$	ϵ_{LCVT}
10	0.1366	61	0.0171	0.0562
20	0.0874	75	0.0158	0.0490
40	0.0919	84	0.0206	0.0448

The convergence of the update algorithm is shown in Fig. 12. The fraction of convergence points changing sign between successive iterations is plotted versus the iteration number. The blue curve consists of the actual values, while the smooth red curve is the fitted exponential curve.

4.1.2. Study of the influence of initial training sample set

The effect of variation of the initial training sample size is studied and documented in Table 1. Selecting a very small initial training set can lead to loss of information in certain regions of the design space. On the contrary, selecting a very large initial set reduces this possibility, but might lead to prohibitive computational times. From Table 1, it is noted that in general the accuracy achieved by the update algorithm is higher compared to an LCVT DOE with the same number of samples.

4.1.3. Study of the influence of stopping criterion

The effect of varying ϵ_1 on the total number of samples is tabulated in Table 2.

4.2. Three-dimensional example with disjoint regions

The decision function for this problem is defined by an analytical function of three variables x_1, x_2 and x_3.

$$f(x_1, x_2, x_3) = \frac{1}{4}(\sin(x_1 - 3)(x_2 - 1) + (x_3 - 1)^2) - 1 \quad (19)$$

The variables x_1, x_2, and x_3 belong to the ranges [0,10], [6,16] and [0,10] respectively.

4.2.1. Construction of the decision function

40 training samples generated using LCVT are used for constructing the initial SVM decision function. The number of convergence points N_{conv} for the stopping criterion is 12,500 and the values of ϵ_1 and ϵ_2 are 1.0×10^{-3} and 5.0×10^{-4} respectively. The number of test points for the error measure is N_{test} = 64,000.

The results are gathered in Table 3. The error for the initial SVM decision function is 14.52%, which is reduced to 2.54% after the update. The total number of training samples needed is 191. In comparison, the error associated with a decision function constructed with 191 LCVT samples is 5.38%. The decision function obtained by SVM before and after the update, starting with 40 samples, are shown in Fig. 13. The decision function obtained by using a 191 point LCVT distribution is also shown in Fig. 14. The SVM decision function and the actual expected function are shown by the light grey and the deep blue surfaces respectively.

Convergence of the update algorithm is shown in Fig. 15. The y-axis represents the fraction of convergence points changing sign between successive iterations and the x-axis represents the

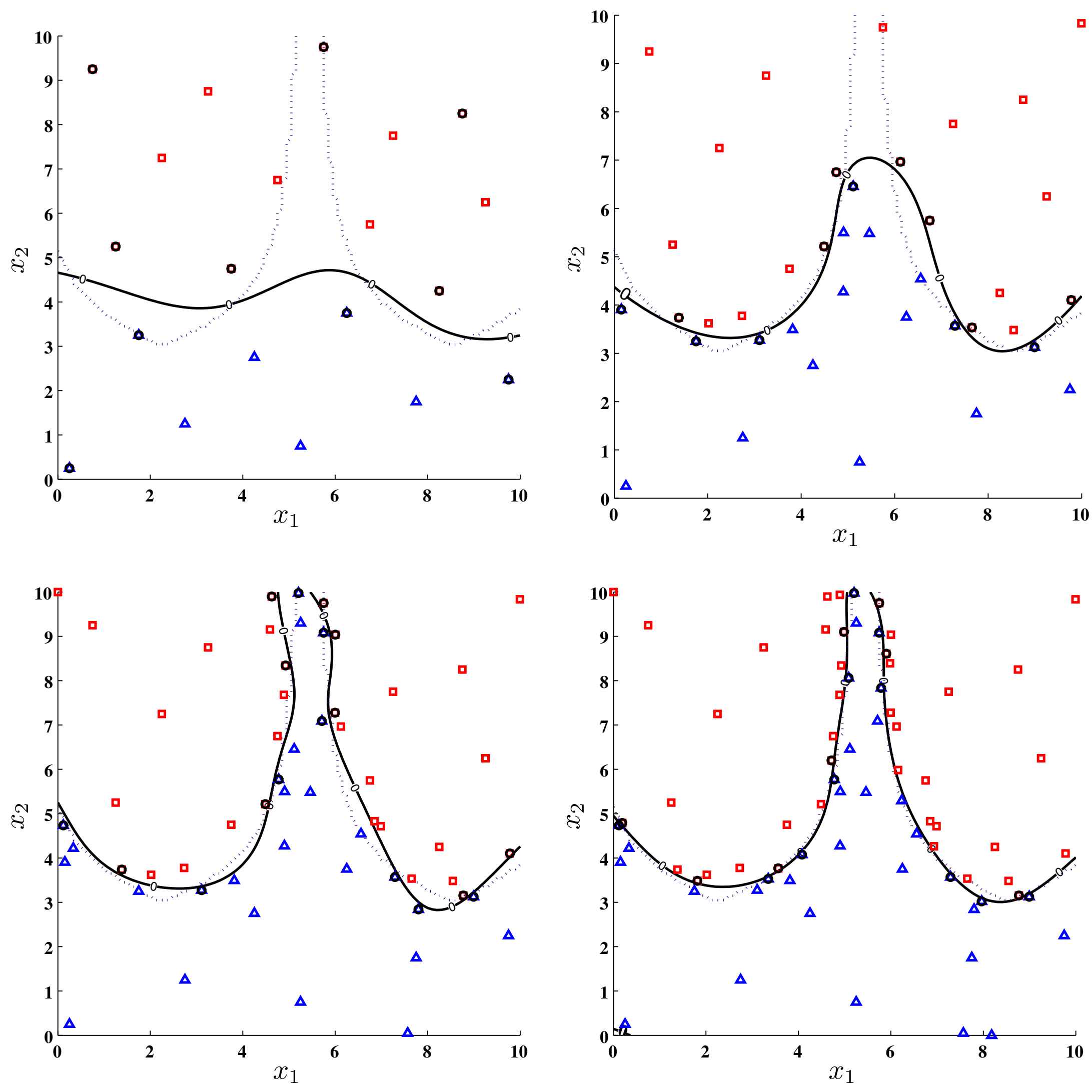

Fig. 10. Two-dimensional problem. Explicit design space decomposition at several stages of the algorithm starting with 20 samples (top left). The final training set size is 75 (bottom right). The SVM and actual decision functions are shown by solid and dotted curves respectively.

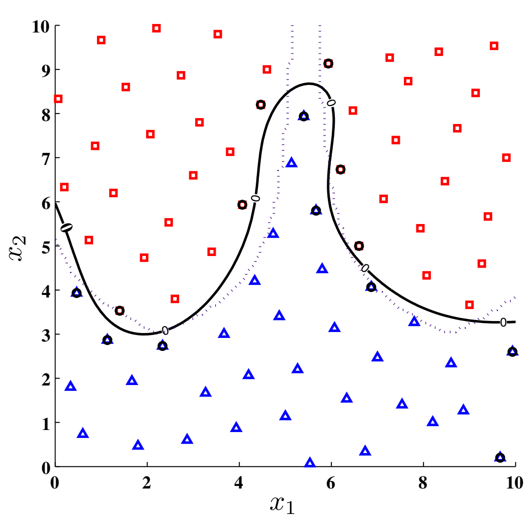

Fig. 11. Two-dimensional problem. Explicit design space decomposition with SVM using 75 LCVT samples. The dotted and solid curves represent the actual and SVM decision functions.

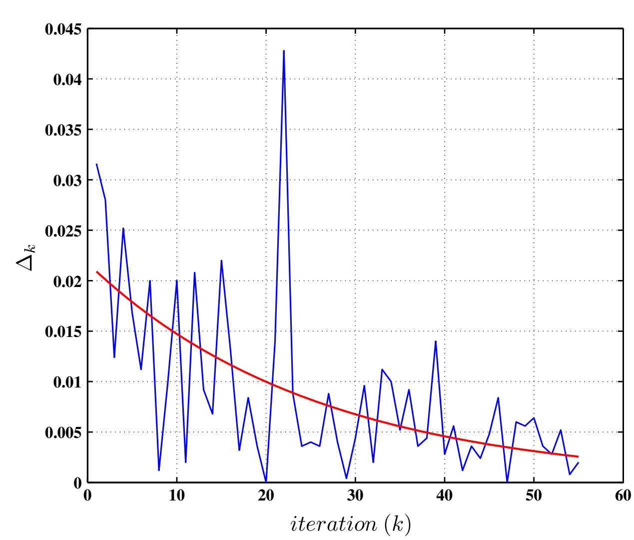

Fig. 12. Two-dimensional problem. Convergence of the update algorithm.

Table 2
Two-dimensional problem. Effect of the value of ϵ_1 for stopping criterion.

ϵ_1	N_{total}	ϵ_{final}	ϵ_{LCVT}
4.0×10^{-3}	75	0.0158	0.0490
3.0×10^{-3}	76	0.0210	0.0398
2.0×10^{-3}	84	0.0153	0.0448
1.0×10^{-3}	117	0.0066	0.0272

Table 3
Three-dimensional problem. Effect of the initial LCVT training set size. The errors are noted for the update scheme and a static LCVT distribution with the same number of samples.

$N_{initial}$	$\epsilon_{initial}$	N_{total}	ϵ_{final}	ϵ_{LCVT}
20	0.1775	174	0.0278	0.0584
40	0.1452	191	0.0254	0.0538
80	0.0765	230	0.0158	0.0394

iteration number. The value of the fitted curve is less than 1.0×10^{-3} at the last iteration.

4.2.2. Study of the influence of initial training sample set

The effect of the variation of initial training set size is studied. The results are given in Table 3. Similar to the two-dimensional problem, it is noted that in general the accuracy achieved by the update algorithm is higher compared to that obtained by a static LCVT DOE of same size.

4.2.3. Study of the influence of stopping criterion

In this section, the number of required samples as a function of ϵ_1 is studied (Table 4).

4.3. Four-dimensional example

For this problem, the decision function is defined by an analytical function of four variables x_1, x_2, x_3 and x_4.

$$f(x_1, x_2, x_3, x_4) = \frac{1}{4}(\sin(x_1 - 3)(x_2 - 1)^2 + (x_3 - 1)x_4) - 3 \quad (20)$$

The variables x_1, x_2, x_3 and x_4 all have range [0,10].

4.3.1. Construction of the decision function

The initial SVM decision function is constructed using 80 training samples generated using LCVT. The number of convergence

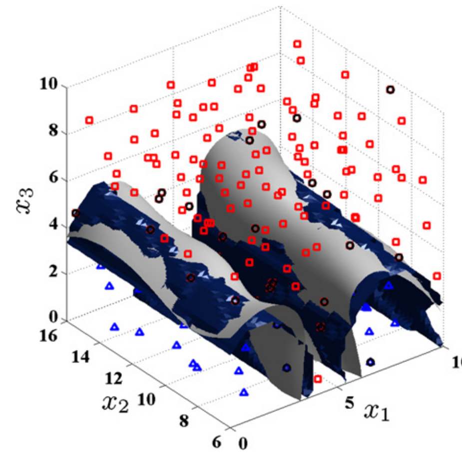

Fig. 14. Three-dimensional problem. SVM decision function generated using 191 LCVT samples. The deep blue surface is the actual decision function and the light grey one is generated by SVM. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

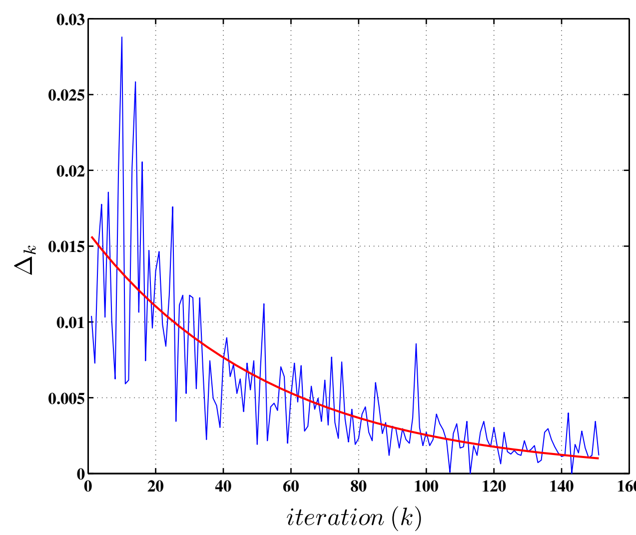

Fig. 15. Three-dimensional problem. Convergence of the update algorithm.

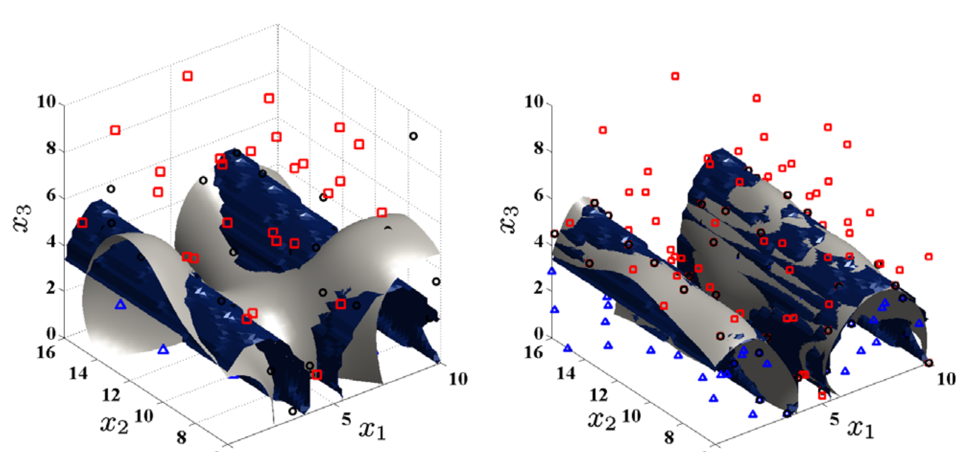

Fig. 13. Three-dimensional problem. The deep blue and light grey surfaces are the actual and SVM decision functions respectively. The left figure shows the initial SVM decision function constructed with 40 LCVT samples while the figure on the right shows the final updated SVM decision function constructed with 191 samples. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Three-dimensional problem. Effect of the value of ϵ_1 for stopping criterion.

ϵ_1	N_{total}	ϵ_{final}	ϵ_{LCVT}
4.0×10^{-3}	125	0.0393	0.0651
3.0×10^{-3}	120	0.0388	0.0647
2.0×10^{-3}	168	0.0262	0.0644
1.0×10^{-3}	191	0.0254	0.0538

Table 5
Four-dimensional problem. Effect of the initial LCVT training set size. The errors are noted for the update scheme and for a static LCVT distribution with the same number of samples.

$N_{initial}$	$\epsilon_{initial}$	N_{total}	ϵ_{final}	ϵ_{LCVT}
40	0.2079	468	0.0443	0.0852
80	0.1687	554	0.0396	0.0798
160	0.1226	629	0.0350	0.0761

points N_{conv} for the stopping criterion is 62,500 and the number of test points for the error measure is N_{test} = 390,625. The values of ϵ_1 and ϵ_2 for the stopping criterion are 1.0×10^{-3} and 5.0×10^{-4} respectively. The error for the initial decision function is 16.87%, which is reduced to 3.96% after the update. The total number of training samples needed is 554. The error associated with decision functions constructed with the same number of LCVT samples is 7.98%. The results are gathered in Table 5.

Convergence of the update algorithm is shown in Fig. 16. The y-axis represents the fraction of convergence points changing sign between successive iterations and the x-axis represents the iteration number. Both the actual Δ_k values and the fitted exponential curve are shown.

4.3.2. Study of the influence of initial training sample set

The influence of varying the initial training set size is studied. The results are given in Table 5. In this case also, it is noted that in general the accuracy achieved by the update algorithm is higher compared to that obtained by a static LCVT DOE of the same size.

4.3.3. Study of the influence of stopping criterion

In this section, the number of required samples as a function of ϵ_1 is studied (Table 6).

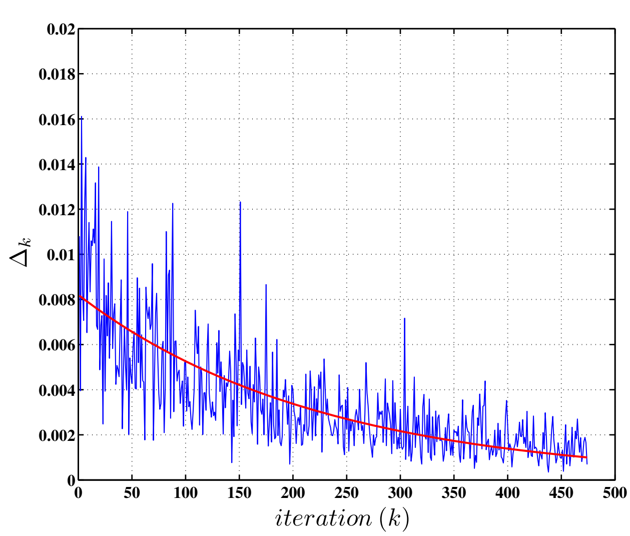

Fig. 16. Four-dimensional problem. Convergence of the update algorithm.

Table 6
Four-dimensional problem. Effect of the value of ϵ_1 for stopping criterion.

ϵ_1	N_{total}	ϵ_{final}	ϵ_{LCVT}
4×10^{-3}	250	0.0841	0.1061
3×10^{-3}	302	0.0705	0.0939
2×10^{-3}	385	0.0536	0.0922
1×10^{-3}	554	0.0396	0.0798

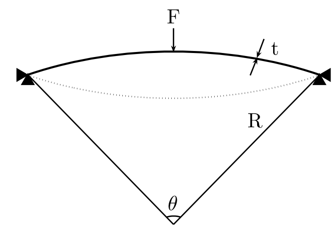

Fig. 17. Arch geometry and loading.

4.4. Arch structure with discontinuous response – construction of explicit decision function

The explicit design space decomposition using SVM is applied to an arch structure subjected to a point load at the center (Fig. 17). The arch is a typical example of a geometrically nonlinear structure exhibiting a snap-through behavior once the limit load is reached. The presence of discontinuities makes the application of response surface methods or other conventional methods difficult or inaccurate. However, the SVM-based method gives an explicit equation of the decision function. The decision function thus obtained, can also be used for the probabilistic optimization of the arch [16,12]. The calculation of the probability of failure using MCS is made efficient, as the explicit equation of the decision function is known in this case.

The arch has a radius of curvature R = 8 m and subtends an angle $\theta = 14°$ at the center of curvature. The thickness t, the width w, and the load F are random variables. The arch structure, simply supported at the ends, is modeled in ANSYS using SHELL63 elements. Due to the symmetries of the problem, only one fourth of the arch needed to be modeled. The range of values allowed for the design parameters are tabulated in Table 7.

To construct the SVM decision function, first an initial LCVT distribution consisting of 10 points is generated with thickness, width and load as the three variables. The variables are normalized by dividing the values by their respective maximum values. The studied response is the displacement of the central node which is solved for at each training sample (design configuration given by the LCVT DOE) using ANSYS. The response shows a clear discontinuity. The discontinuous variation of the displacement with respect to the thickness and width is depicted in Fig. 18 for a fixed value of the applied load.

Table 7
Range of design parameters for arch problem

	Thickness (t)	Width (w)	Force (F)
Min value	3 mm	150 mm	2000 N
Max value	10 mm	500 mm	8000 N

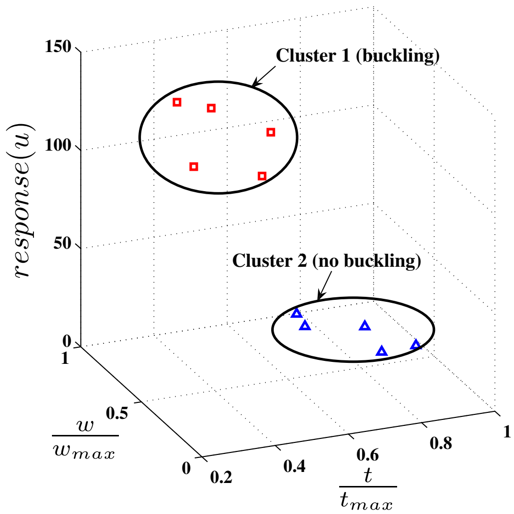

Fig. 18. Discontinuous response of arch. The response (displacement) is obtained for a constant load $F = 6400$ N.

The discontinuity in displacement is used to separate the responses into two clusters using K-means clustering. One of the clusters corresponds to buckling (failure) while the other corresponds to design configurations which do not exhibit buckling. These two classes of samples in the design space are labeled as "+1" and "−1". This information is input to the SVM algorithm to create the initial decision function. Once the initial SVM decision function is obtained, it is adaptively updated using the aforementioned algorithm. At every iteration the displacement of the new point is solved for. The new sample is added to the training set, and K-means clustering is then used again to reassign class labels to all the training samples based on their respective displacement values. After reassigning the class labels, SVM is reconstructed. The information is used for the selection of a new training sample in the next iteration, until the stopping criterion is met.

The number of convergence points N_{conv} for the stopping criterion is 312,500, and the values of ϵ_1 and ϵ_2 are 1.0×10^{-3} and 5.0×10^{-4} respectively. The number of training samples required to construct the final updated SVM decision function is 48. The initial and final SVM decision functions are shown in Fig. 19. For comparison, an SVM decision function is also constructed using 48

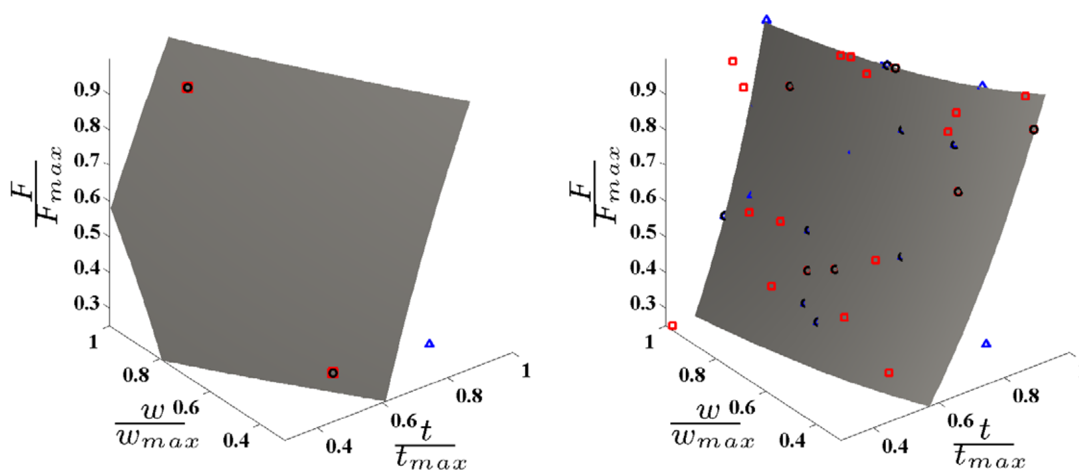

Fig. 19. Arch problem. The left and right figures show the initial and final updated SVM decision functions constructed with 10 and 48 samples respectively.

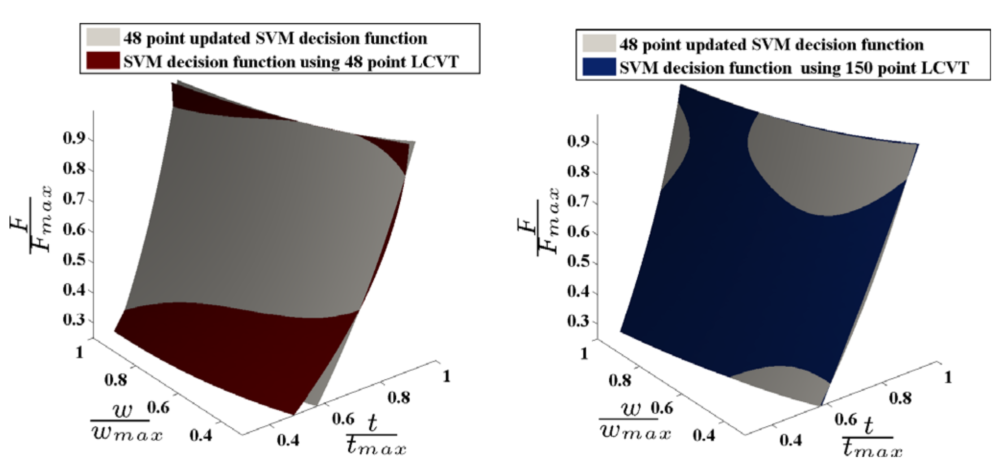

Fig. 20. Three-dimensional arch problem. Comparison of SVM decision functions constructed using update algorithm and otherwise. The dark brown and light grey surfaces in the left figure are the decision functions using 48 LCVT samples and the update algorithm respectively. The deep blue surface in the right figure is the decision function constructed with 150 LCVT samples. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

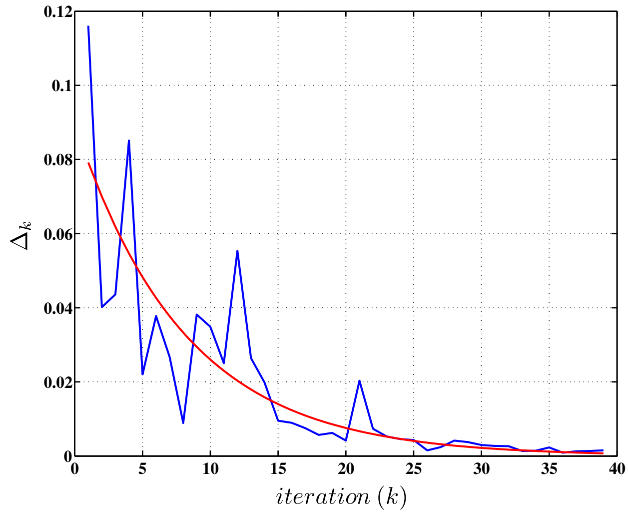

Fig. 21. Arch problem. Convergence of the update algorithm.

LCVT training samples. Fig. 20 shows that the decision function generated using 48 LCVT samples (dark brown surface) deviates from the updated SVM decision function (light grey surface). On the contrary, the updated decision function is very similar to the decision function (deep blue surface) constructed with a larger LCVT training set of 150 samples.

Convergence of the update algorithm is shown in Fig. 21. The fraction of convergence points changing sign between successive iterations is plotted against the iteration number. Both the actual Δ_k values, and the fitted exponential curve are shown.

5. Concluding remarks

5.1. Summary

An approach to adaptively update explicit decision functions constructed with SVM is proposed. The technique provides an efficient sampling strategy as it only uses relevant samples. It is therefore of interest for problems involving high computational times. In addition, a major strength of the SVM-based explicit design space decomposition lies in its ability to handle discontinuous responses.

A general stopping criterion based on the variations of the predicted decision functions is described, thus providing an approach to automatically find the number of required training samples. The efficiency of the methodology is demonstrated through its application to various test problems. An error measure was also developed in the case of analytical test examples.

5.2. Discussion and future work

The proposed methodology could benefit from several incremental improvements that are discussed below:

- The next stages of this research involve the application of the scheme to more complex practical engineering problems involving more variables. However, in the case of computationally intensive function evaluations (e.g., a nonlinear transient finite element simulation), the objective is to accurately solve problems with 10–15 variables.
- The minimum distance between samples which is defined as a function of the hypervolume of the design space, the problem dimensionality, and the number of training samples, consists of a constant coefficient. The effect of the value of the coefficient needs to be studied more rigorously. The efficiency of the update algorithm might be improved by updating the coefficient during the course of the construction of the decision function.
- In the present approach, the convergence criterion is based on a large number of convergence points. It is therefore, to be accurate, limited to a handful of dimensions. Future research will involve the development of an alternate convergence criterion.

In addition, the approach is suitable for cases where the state of a system (e.g., failure or safe) cannot be assessed by comparing a response to a threshold. That is, the decision function and its update could be constructed from qualitative experimental data only or combined with simulation results.

Appendix A. Study of the relation between accuracy and the total number of training samples

In order to demonstrate the relation between the update strategy and the quality of the decision function, the number of training samples required to achieve a given accuracy is studied. The results, though generic, are shown for a particular two-dimensional analytical function representing disjoint regions:

$$f(x_1, x_2) = x_2 - x_1 \sin(\mu x_1 + 2) - 3 \quad (21)$$

The variables x_1 and x_2 are considered as uniformly distributed, both having range [0,10]. The region where $f(x_1,x_2) > 0$ is labeled +1 and the complementary region is labeled -1. The number of training samples required to achieve specific levels of accuracy are listed in Tables 8 and 9 for $\mu = 1$ and $\mu = 1.5$ respectively. The total samples required for successive increments in accuracy by a factor of 2 are noted in both cases. When a stopping criterion with $\epsilon_1 = 1.0 \times 10^{-3}$ and $\epsilon_2 = 5.0 \times 10^{-4}$ is used, a final error of 2.45% is obtained with $\mu = 1$. For $\mu = 1.5$, a final error of 4.58% with 140 training samples is attained using the same stopping criterion.

It is seen that the rate of increase in accuracy is high in the beginning, and reduces gradually. The stopping criterion is a trade off between high accuracy and computational cost (i.e., number of samples), and depends on the requirements of a specific problem. The visual representations of the decision functions with 1% and 2% error for $\mu = 1$ and $\mu = 1.5$, respectively are depicted in Fig. 22.

Table 8
Study of increase in the accuracy with number of training samples ($\mu = 1$)

ϵ	N_{total}
12.71	20
6.12	35
3.00	91
1.50	106
1.00	118

Table 9
Study of increase in the accuracy with number of training samples ($\mu = 1.5$)

ϵ	N_{total}
19.62	20
9.90	61
5.00	124
2.50	232
2.00	253

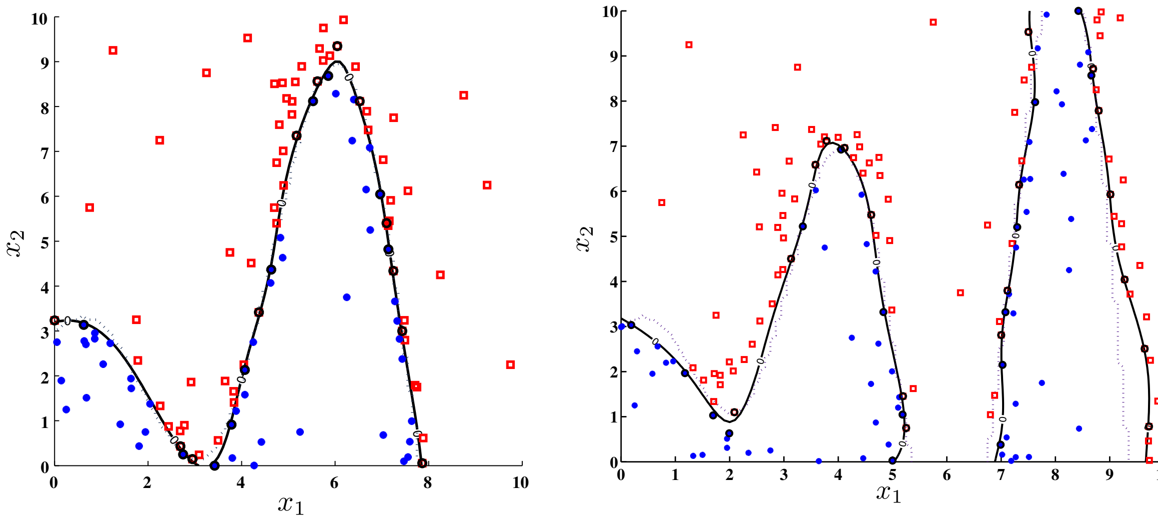

Fig. 22. Final decision functions with 1% and 2% error for $\mu = 1$ and $\mu = 1.5$ respectively. The solid curves represent the predicted SVM decision functions and the dotted curves show the actual decision functions.

References

[1] Myers RH, Montgomery DC. Response surface methodology. 2nd ed. Wiley; 2002.
[2] Melchers R. Structural reliability analysis and prediction. John Wiley and Sons; 1999.
[3] Montgomery DC. Design and analysis of experiments. Wiley and Sons; 2005.
[4] Wang GG, Wang L, Shan S. Reliability assessment using discriminative sampling and metamodeling. SAE Transactions. J. Passenger Cars: Mech Syst 2005;114:291–300.
[5] Huang D, Allen T, Notz W, Miller R. Sequential Kriging optimization using multiple-fidelity evaluations. Struct Multidisciplinary Optim 2006;32:369–82.
[6] Bichon BJ, Eldred MS, Swiler LP, Mahadevan S, McFarland JM. Multimodal reliability assessment for complex engineering applications using efficient global optimization. In: Proceedings of the 48th conference AIAA/ASME/ASCE/AHS/ASC on structures, dynamics and materials. Paper AIAA-2007-1946, Honolulu, Hawaii; April 2007.
[7] Hurtado Jorge E. An examination of methods for approximating implicit limit-state functions from the viewpoint of statistical learning theory. Struct Safety 2004;26(July):271–93.
[8] Martinez WL, Martinez AR. Exploratory data analysis with matlab. Comput Sci Data Anal Ser. Chapman and Hall/CRC; 2005.
[9] Haldar A, Mahadevan S. Probability, reliability, and statistical methods in engineering design. New York: Wiley and Sons; 2000.
[10] Youn BD, Choi KK, Du L. Adaptive probability analysis using an enhanced hybrid mean value method. Struct Multidisciplinary Optim 2005;29(2):134–48.
[11] Missoum S. Controlling structural failure modes during an impact in the presence of uncertainties. Struct Multidisciplinary Optim 2007;34(6):463–72.
[12] Missoum S, Benchaabane S, Sudret B. Handling birfurcations for the optimal design of transient dynamics problems. In: Proceedings of the 45th conference AIAA/ASME/ASCE/AHS/ASC on structures, dynamics and materials. Paper AIAA-2004-2035, Palm Springs (CA), USA; April 2004.
[13] Missoum S, Ramu P, Haftka RT. A convex hull approach for the reliability-based design of nonlinear transient dynamic problems. Comput Meth Appl Mech Eng 2007;196:2895–906.
[14] Shawe-Taylor J, Cristianini N. Kernel methods for pattern analysis. Cambridge University Press; 2004.
[15] Tou JT, Gonzalez RC. Pattern recognition principles. Addison-Wesley; 1974.
[16] Basudhar A, Missoum S, Harrison Sanchez A. Limit state function identification using support vector machines for discontinuous responses and disjoint failure domains. Probabilistic Eng Mech 2008;23(1):1–11.
[17] Beachkofski BK, Grandhi R. Improved distributed hypercube sampling. In: Proceedings of the 43rd conference AIAA/ASME/ASCE/AHS/ASC on structures, dynamics and materials. Paper AIAA-2002-1274, Denver (CO), USA; April 2002.
[18] Romero VJ, Burkardt JV, Gunzburger MD, Peterson JS. Comparison of pure and Latinized centroidal Voronoi tesselation against various other statistical sampling methods. J Reliab Eng Syst Safety 2006;91:1266–80.
[19] Greg Schohn, David Cohn. Less is More: Active learning with support vector machines. In: Proceedings of the 17th international conference on machine learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2000. p. 839–46.
[20] Clarke SM, Griebsch JH, Simpson TW. Analysis of support vector regression for approximation of complex engineering analyses. J Mech Design Trans ASME 2005;127(November):1077–87.
[21] Gunn SR. Support vector machines for classification and regression. Technical Report ISIS-1-98, Department of Electronics and Computer Science, University of Southampton; 1998.
[22] Giunta AA, Wojtkiewicz SF, Eldred MS. Overview of modern design of experiments methods for computational simulations. In: 41st AIAA Aerospace Sciences Meeting and Exhibit. Paper AIAA-2003-649, Reno (NV); 2003.
[23] Butler AN. Optimal and orthogonal latin hypercube designs for computer experiments. Biometrika 2001;88(3):847–57.
[24] Liefvendahl M, Stocki R. A study on algorithms for optimization of latin hypercubes. J Statis Plann Inference 2006;136:3231–47.
[25] Hartigan JA, Wong MA. A K-means clustering algorithm. Appl Statis 1979;28:100–8.
[26] Hamerly G, Elkan C. Alternatives to the K-means algorithm that find better clusterings. In: 11th International conference on information and knowledge management (CIKM'02); November 2002. p. 600–7.